D[illegible] A's COPY

ARCNET Factory LAN Primer

Contemporary Control Systems, Inc.
Downers Grove, Illinois USA

Second Printing

ARCNET Factory LAN Primer
Contemporary Control Systems, Inc.
2500 Wisconsin Avenue
Downers Grove, Illinois USA 60515
(312) 963-7070

Co-authored by: Katherine S. Hoswell and George M. Thomas
CCSI Part Number: SD100000-00B

First Printing, November 1987
Second Printing, April 1988

Acknowledgments

To Datapoint Corporation; Standard Microsystems Corporation; NCR Corporation, Microelectronics Division; Performance Technology, Inc.; The ARCNET Trade Association; Pro-Log Corporation, Raycom Systems and *Control Engineering* for all the assistance in developing this book.

Trademarks

This primer was developed in-house using WordStar Professional and Xerox Ventura Publisher on an IBM AT and Apple's Laserwriter. The graphics were accomplished using Autodesk's AutoCAD also on an IBM AT.

CONTENTS

PREFACE

American industry is in the midst of a revolution, one that is being touted as the salvation for our eroding manufacturing base. This revolution is computer integrated manufacturing (CIM) and it blends together the use of automated equipment with a redirection in manufacturing philosophy that focuses primarily on issues such as commitment to quality, just-in-time production, small lot sizes, and a paperless factory. Implementing CIM in America's factories will require a great deal of commitment both on the part of management and the work force, for retooling and retraining will be extensive.

The key to making CIM work is full integration of all functions within a company from the office to the factory floor. Invariably, this integration will require the use of local area networks, or LANs, that can link various automated devices manufactured by multiple vendors.

Contemporary Control Systems, Inc. (CCSI) has been in the LAN business for the last five years. The company was originally founded in 1975 to design and manufacture industrial control systems, incorporating the newly-introduced microprocessor into industrial control systems. Recognizing the growing need to network these systems, CCSI entered the local area network market in 1982 and has since worked with one LAN in particular—ARCNET. ARCNET was developed by Datapoint Corporation of San Antonio, Texas. CCSI first worked with ARCNET on a client application which linked distributed microcomputers in a newspaper front-end editing sys-

tem. ARCNET provided exceptional reliability and high performance in a non-stop application. CCSI recognized that these same virtues of ARCNET were applicable to the industrial market which also requires communications that provide performance, low cost, high reliability, easy expansion and operational simplicity. ARCNET possesses all of these qualities but few people know of its existence.

To popularize ARCNET and its strengths for industrial applications, CCSI wrote this primer which explains the intricacies of ARCNET in both layman and technical terms. Section I of this book covers basic networking concepts. Section II expands into greater detail on both the hardware and software aspects of ARCNET. Most of the technical information was derived from publications from either Datapoint Corporation, Standard Microsystems Corporation or NCR Corporation, Microelectronics Division. Credit is freely given to the early pioneers of ARCNET.

Several examples of both ARCNET hardware and software are given in the primer. CCSI's products are used as examples since those products are most familiar to CCSI. It should not be construed that they are the only products that will perform those functions. ARCNET is supported on many microprocessor bus structures and the use of ARCNET is expanding. CIM is not going to be implemented by only one vendor, but by many. Integrating the factory requires communication networks that link various vendors. ARCNET is one possible solution.

SECTION ONE:

INTRODUCTION TO LOCAL AREA NETWORKS AND ARCNET

1 THE STANDARDIZATION MOVEMENT

There is a growing trend in the factory communications industry to implement standards for integrating automated equipment that provides freedom from vendor dependency and protects users from changes in technology. This trend has its roots in the MAP (manufacturing automation protocol) movement. The goal of MAP is to define a network communications structure for multivendor factory automation systems. By having a specified communications scheme supported by many vendors, users can configure systems with "off-the-shelf" equipment.

The initiative for MAP originated from General Motors Corporation. GM's plants have many "islands of automation" that are usually equipped by a single vendor. Generally, the equipment within an island can communicate; however, problems arise in establishing communications between "islands" since it is rare that one vendor can supply all of the automation equipment. With this lack of integration, GM's plants have become a maze of wires, custom interfaces and protocols. The MAP movement is GM's attempt at developing a standardized LAN to reduce the confusion and establish a fully-integrated system.

The movement to develop communications standards is a positive move for the factory communications industry, especially since this technology is still in its infancy. However, the MAP movement has been slow to evolve and, by the looks of

the new 3 1/4" thick 3.0 MAP Spec, extremely complex. Additionally, MAP's cost per node is very high. Another standard that is evolving alongside MAP is Boeing's TOP—Technical Office Protocol—which links engineering and office functions and will eventually bridge to MAP. A bridge will have to be developed since these two networks are designed with inherently incompatible access methods.

DEMAND FOR MULTIVENDOR SOLUTIONS

Users need a multivendor networking solution today that can be implemented at a reasonable cost. While the MAP effort is noble, it is not the only choice available. Not all applications require the complexity of MAP. Other multivendor networking systems exist that have become de facto standards through their extensive use, such as ARCNET. This LAN has proven its worth in both the office and factory environment with over 600,000 installed nodes worldwide; and ARCNET requires no bridging between the office and factory functions. ARCNET's token-passing protocol uses the same access method chosen for MAP and has demonstrated remarkable reliability in real-time production environments. But most important, ARCNET is a mature technology that is inexpensive to implement, easy to understand and use, and available now.

Companies must analyze their needs and determine whether or not non-MAP systems can meet their goals. ARCNET should be considered as another networking alternative to MAP, not that it can replace MAP, but ARCNET can meet the needs of most networking applications. If the user decides to convert to MAP at a later date, ARCNET can function as a subnetwork and be bridged to MAP. In fact, at least one vendor

has successfully implemented ARCNET on the same broadband communications cable specified by MAP.

In terms of ARCNET's level of support for the International Standards Organization's Open Systems Integration Model (OSI), another standard for networking systems endorsed by MAP, ARCNET presently conforms to the model up through level two and part of level three. Efforts have been initiated among members of the ARCNET community to develop software standards for levels 3, 4, 5 and 6. Through ARCNET's growing use and adoption, the LAN is moving away from being strictly a proprietary network and is evolving into more of an open architecture system. There is even some discussion among ARCNET vendors of achieving IEEE certification by developing an 802.X specification. Two companies manufacture ARCNET chip sets using different silicon technologies; however, both chip sets are compatible which gives the industry two chip sources. Using the same chip set ensures hardware compliance among all ARCNET vendors and removes compatibility concerns.

The race to interconnect factory equipment is just beginning and more options for achieving this goal are becoming available everyday. Many companies cannot afford to sit back and wait for an evolving standard to actually become accessible and affordable. For these companies, ARCNET may be today's answer.

2 LAN BASICS

Traditionally, the data processing department in a manufacturing company has determined the computer needs of the office environment. The use of automation on the factory floor was spotty, some companies had state-of-the-art automated production tools while others had crude forms of automated equipment. Selecting computers for the factory was outside the domain of the DP department as long as the computer was called a "controller."

With the advent of the microcomputer, automation efforts became confused and companies found themselves owning hundreds of incompatible devices. Departments purchased computers for their own needs without giving any consideration to eventually integrating these computers with other departments. Today, these companies are realizing that to be truly successful they must link every department both in the office and the factory to form one system.

Local area networks are becoming the most popular means of "closing the complete factory loop" in a simple, cost-effective way. LANs connect all types of computerized devices throughout a company enabling users to not only share data but other resources such as printers and data storage devices without having to rely upon one large high-speed computer.

LAN APPLICATIONS

LANs have far-reaching potential for enhancing the speed and flexibility at which data is transferred on the factory floor, thus improving overall manufacturing performance. Consider, for example, a chemical company that has multiple mixing tanks, each batching different recipes 24 hours a day.

With a LAN, a central computer can be linked to distributed programmable logic controllers (PLCs) at each mixing tank to develop an integrated processing control system. The PLCs control the mixing process, constantly metering and controlling several variables in the mixing tank. The central computer stores the recipes for each of the PLCs and downloads the proper recipe for each tank while at the same time monitoring and recording production from each tank. All data communications occur over the same local area network.

LANs are also finding use in robotics applications, for example, networking a line of robots in an automobile manufacturing plant. The LAN could link the robots to a supervisory system which stores programs for the robots and prints out error messages occurring at robot stations.

This is now being done at Volvo's factory in Gothenburg, Sweden, using an ARCNET LAN and CIMNET software which was developed by Comendec, a British company that designs industrial network systems. The supervisory system is centrally controlled by a Digital Equipment Corporation

LSI 11/73 minicomputer which provides data storage, a CRT, two printers and a tape spooler. The system monitors 200 Cincinnati Milacron T3-586 robots used for welding. Via the LAN, individual tape records are loaded automatically into the robots.

Before the LAN was implemented, robots were reprogrammed individually, a half hour task per robot. Over the ARCNET LAN, the CIMNET software loads programs into the robots at speeds of 15 to 20 seconds per robot which translates into a total of about 20 minutes for programming all 200 robots. Not only does this LAN save time, but it also eliminates costly human programming errors.

WHAT IS A LAN?

Hundreds of definitions exist for describing LANs and their components. In this primer, we define a local area network as a combination of hardware and software that enables two or more computerized devices to share data base information, hardware resources and other application programs. A LAN also eases the task of connecting nodes on the network as well as freeing up access to any resource on the network.

Unlike a wide area network, like the telephone system, a LAN is owned and maintained by the user and is usually less than a mile in total distance (although LANs can reach distances of 50 miles).

A LAN is a data communications system but it is unlike a typical point-to-point link such as an RS-232-C connection—an Electronic Industries Association standard. The RS-232-C link primarily defines the interface between data terminal equipment and data communications equipment. Although such a system is relatively inexpensive to implement, it's extremely limited in function and is much slower than a LAN. Data rates on an RS-232 link usually do not exceed 19,200 baud, whereas LANs can transmit data at 1 to 10 megabits/sec. LANs offer a high level of flexibility in the kind of devices that can be linked and how they are connected. LANs also allow greater distances between nodes.

There are several kinds of local area networks, each offering benefits to the user depending on the application. In this primer, we will focus on the ARCNET LAN. Before we get into the specifics of ARCNET; however, we will quickly cover the basics of LAN systems.

LAN HARDWARE

Networks are comprised of both hardware and software. The hardware components can be categorized into four main areas: topology, transmission medium, signaling method and access method.

Topology

Topology is the actual layout of the cables connecting the nodes to the LAN. The three most common topologies are bus, star and ring configurations.

Bus Topology

A bus topolgy, shown in Figure 1, is a traditional multidrop configuration that connects each node to a single cable, or bus, via a line tap. On a bus topology, nodes do not forward messages. Every node hears all the transmissions on the bus because the bus serves as a broadcast medium.

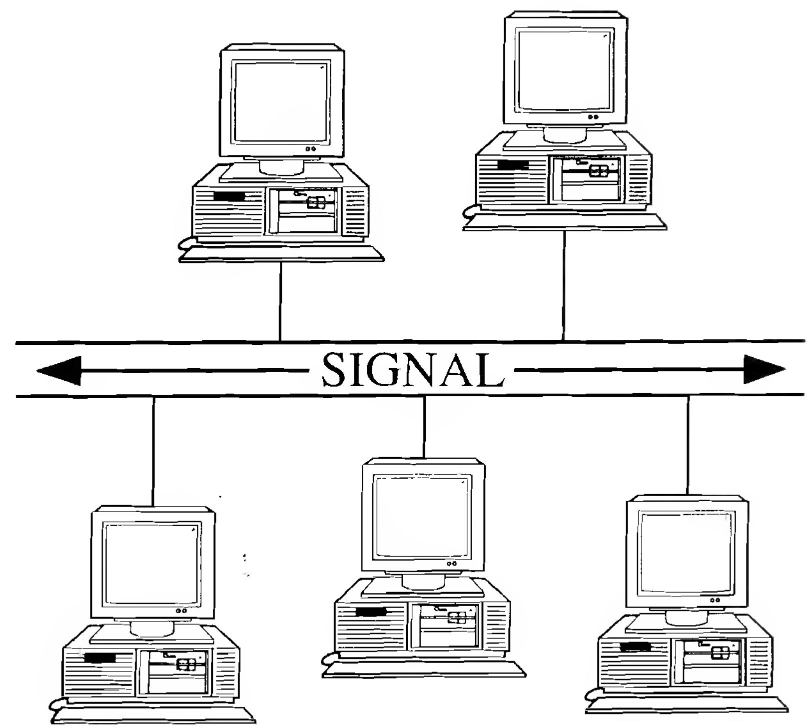

Figure 1. A Bus Topology

Star Topology

The physical layout of a star topology looks somewhat like a starfish—all nodes are wired to a central station or hub that provides cabling access for each node, see Figure 2 below. Nodes can be added to the network by linking them to the central controller. However, once all of the ports on the central controller are in use, no extra nodes can be added.

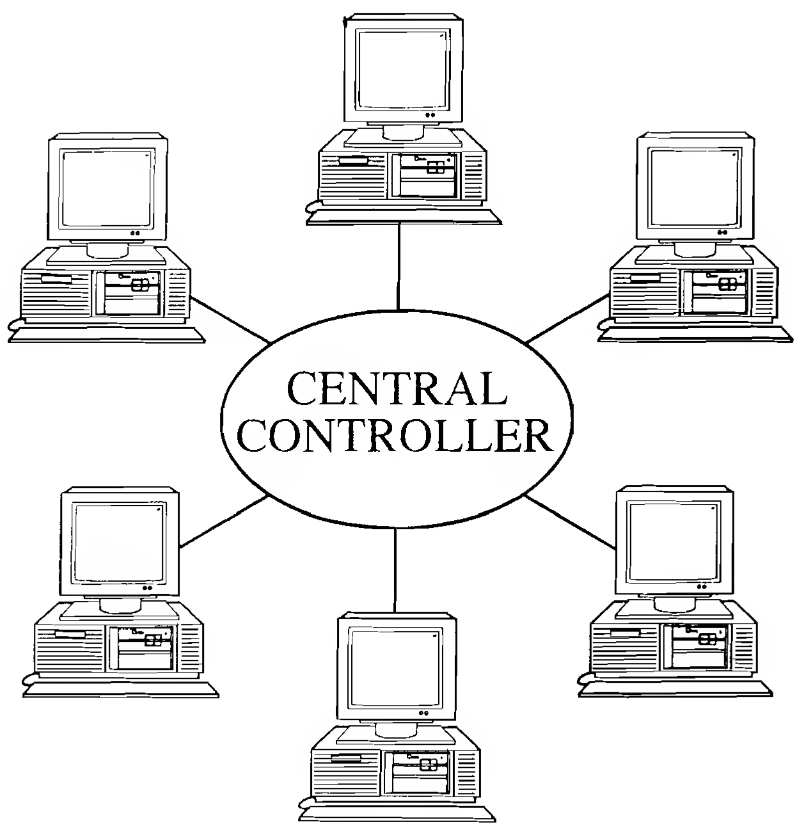

Figure 2. A Star Topology

Ring Topology

On a ring topology, nodes connect to one another forming one contiguous loop as shown in Figure 3 below. Data passes from node to node, always in the same direction.

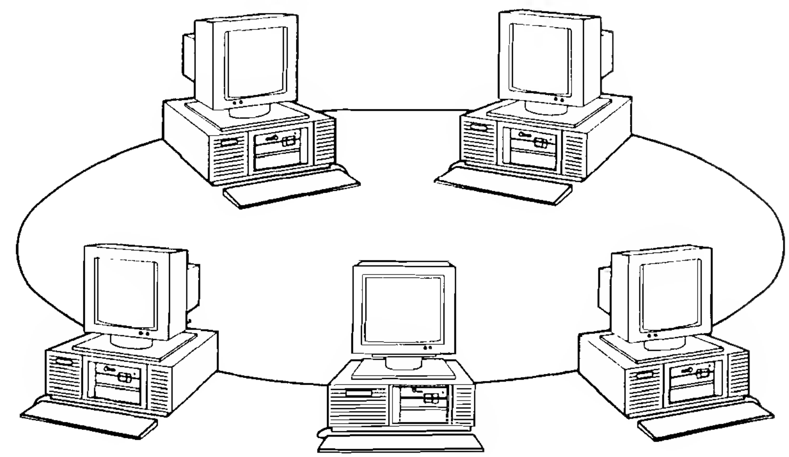

Figure 3. A Ring Topology

Transmission Medium

The media chosen for a LAN is very important. Each media has advantages for specific applications. When choosing a LAN, it is best to find one that supports multiple media which provides the ultimate in flexibility. In general, there are three choices for transmission media: coaxial cable, twisted pair and fiber optic cable.

Coaxial Cabling

Coaxial cable is the most popular cable used today, known primarily because of its widespread use in the cable television industry. Coaxial cable has a shield that protects signals from outside noise and interference, especially apparent in manufacturing environments. This cable is recommended for moderate to high speed networks passing data between 1 to 15 megabits/second.

Twisted Pair Cabling

On the other hand, twisted pair cable offers minimal protection from noise since the cable is not shielded. Twisted pair cable is easy and inexpensive to install, but it can only handle low transmission data speeds, typically up to 2 megabits/sec. New twisted pair cabling is now being introduced that can support higher data rates, but the noise interference problems are yet to be solved and much attention must be given to adjacent cabling.

Fiber Optic Cabling

Fiber optic cabling offers the maximum protection from electrical noise. This medium offers high speed transmission rates up to 200 megabits/second. Fiber optics is also approved for use in hazardous locations such as explosive manufacturing environments since light travels through the cable rather than electrical signals. While fiber optics costs far more than the other two types of media, it can cover larger distances and provide exceptional data security.

Signaling Method

Data is transmitted using basically three different schemes—baseband, broadband and carrierband. Choosing the signaling method for a LAN is a significant decision because these networks vary considerably in flexibility, expandability, cost of installation, ease of installation, and volume of data traffic they can handle.

Baseband Networks

Baseband networks transmit either analog or digital signals over the transmission medium, one message at a time, without modulating a carrier. In an ARCNET system, the digital signal is encoded into a dipulse by the cable transceiver before entering the transmission medium. Upon reaching their destination, the signals are decoded back to their digital form.

In general, baseband networks are far less costly to implement than broadband systems. In a baseband system, the hardware requirements include cabling, network interface modules, and electronic repeaters to amplify and retransmit signals. On the other hand, broadband networks require: amplifiers, attenuators, terminators, remodulators, splitters, RF modems, higher cost cable, and attention to engineering detail for implementation.

Baseband networks are flexible to install and are easily expanded. In fact, almost anyone can install a baseband system given the right components. It's simply a matter of connecting the cable to the network nodes. With active hubs, baseband

networks can be tapped into almost anywhere with virtually no engineering required.

Broadband Networks

Broadband networks can transmit both digital and analog signals such as data, voice or video (most commonly used for security applications) by sending data over separate, dedicated channels on the network. The network medium is basically a backbone running through the center of the plant, usually in the ceiling, with taps dropping down at points of connection. Before signals enter the transmission medium, the signal is modulated into noninterfering frequencies by a radio frequency (RF) modem. Upon reaching the destination node, signal frequencies are demodulated back into their original form by another RF modem. By modulating signals over the network, multiple networks such as security, telephone and data processing systems can coexist on the same cable with each "network" running on its own channel.

Although broadband offers flexibility in signal transmission, broadband costs are much greater than baseband due to the hardware needed and the engineering expertise required to design a system with a proper balance of signal strength across the frequency band.

Once a broadband implementation is complete, nodes cannot be added to the network beyond the original allocation of taps without re-engineering that section of the network—severely limiting the expandability of broadband networks. However, for large installations requiring the use of multiple channels over one cable, broadband is the only choice.

Carrierband Networks

Carrierband differs from baseband in that signals are modulated before transmission. Unlike a multiple-channel broadband network, signals travel over only one frequency channel in a carrierband network—a carrier channel can be one of the channels on a broadband network. In terms of hardware, carrierband networks require modems; however, head-end remodulators (required in broadband) are not needed.

Access Method

The access method is the protocol or set of rules that governs how data is actually transmitted over the network. In other words, the access method defines how each node on the network attains access to the network medium to transmit data and how each receiving node receives the data.

Polling Method

Basically, two types of access methods exist for local area networks—polling and contention. In a polling system, each node on the network is "asked" if it has anything to transmit. In a contention network, each node must compete for access to the network.

The two most dominant access protocols used today are token passing—a polling system, and carrier-sense multiple access with collision detection (CSMA/CD, IEEE Standard 802.3)—a contention system.

With the token passing protocol, each node is polled via a token (a short data signal), to determine if the node has a data

packet to transmit. If it does, it becomes the momentary master of the network and transmits its message upon reception of the token. If the node has no data to transmit, it releases the token. Every node is assured access to the network within a certain determinable amount of time since the maximum length of the packet is defined.

Contention Method

On a contention system, nodes compete for access to the network. A contention system operates similarly to a conversation between two people. One person talks while the other listens. When the first person pauses, the second person talks. If both talk at once, both pause for a random amount of time, and the first person to begin talking without interruption is the one that controls the conversation. When more than two people are involved in a conversation, getting a chance to talk becomes more difficult. When more people are involved, there is a greater chance of two talking at once, or as in a contention system, two messages colliding during transmission which requires retransmission of both messages.

A disadvantage of a contention system is that no opportunity exists for calculating the worst case time for a node to transmit its message over the network. Also, collisions can occur on a contention system, requiring retransmission. When a collision occurs, both nodes sense the collision and each waits a random period of time before retransmitting the message. If another collision occurs, each node must retransmit once more until the messages are sent. The probability of collisions occurring increases as nodes are added to the network. Depending upon the number of network nodes, the frequency of transmissions

over the network, and the application, a CSMA/CD system may be impractical for real-time computing environments such as those frequently found in factories.

LAN SOFTWARE

Software also plays an important role in designing an efficient LAN. Connecting the network devices with cables doesn't automatically establish communications. There must be a means of controlling the flow of data over the network.

In a LAN, two types of software are needed—software drivers and network operating system software. The software drivers control the passage of data over the network, providing the actual set of instructions that corresponds to the access method used in the network. Sample software drivers are included in Appendix A of this book.

The network operating system software is far more complex than the software drivers. The primary function of network operating system software is to service user requests for network resources such as data files and access to printers on the network. Basically, two types of network operating systems software exist—shared resource and peer-to-peer.

Shared Resource Systems

In a shared resource environment, a central file server handles the distribution of all files on the network. The file server stores network routines and high-speed disk access routines. Software also exists on each node to intercept any peripheral request and reroute the request to the file server if necessary.

Peer-To-Peer Networks

In a peer-to-peer network, the software resides on each node and like the shared resource environment, the software intercepts peripheral requests and redirects them to other nodes on the network, when needed. Unlike a shared resource system, there is no centralized file storage and the nodes are not optimized for file access, resulting in lower performance than a shared resource network.

Popular Network Software Packages

Today, there are several vendors of network operating software packages that support a variety of networking protocols. The network operating systems that support ARCNET include: Novell, Inc.'s Advanced NetWare, Western Digital's ViaNet, Banyan System's Vines, and Torus' Tapestry.

Of the two major types of network operating system software, shared resource versus peer-to-peer, the two more popular systems are Advanced NetWare and Vianet, respectively.

Advanced NetWare, probably the most commonly used system with any of today's LANs, utilizes a central file server which handles all requests for data files and sharing of peripheral devices such as printers. With NetWare, all nodes on the network must interface with the file server (at this time, NetWare does not support peer-to-peer communications). This configuration offers fast access to data files as well as high data security and centralized file backups.

In December, 1986, an independent benchmark test was performed on six LANs running under Novell's NetWare—

Standard Microsystems Corp.'s ARCNET, Texas Instrument's Business-ProLAN Server, Gateway Communication's G/NET, Novell's NetWare/S-Net, Proteon's ProNET-10 and AT&T's StarLAN. PC Magazine conducted the test and rated ARCNET as the editor's choice for three main reasons: its ease of installation, resistance to degradation under heavy loads, and its error-checking and retransmission protocol. It is interesting to note that although ARCNET is a lower speed LAN, as compared with some of its competition, it still exhibited exceptional throughput.

ViaNet differs from NetWare primarily in its approach to data storage and access. Vianet is a peer-to-peer network operating system where each node is connected to the network transmission media without requiring the use of a file server. When a user requests a data file over the network, the user must specify the file name and the node which is storing the data, since there isn't a central file server storing users' data.

Choosing a network operating system can be the most important decision in choosing a local area network. As with most computer systems, if the software is slow, the system suffers regardless of how fast the hardware runs.

References

Bartram, Colin. "An ARCNET System Links Multivendor Robots." *Control Engineering*, October 1987, 2nd edition, pp. 37—38.

Derfler Jr., Frank J. "Making Connections: LANs Under NetWare." *PC Magazine*, December 9, 1986, pp. 149—191.

McNamara, John A. *Local Area Networks An Introduction to the Technology*, Burlington, MA: Digital Press, 1985.

Stacy, Alan H. *The MAP Book An Introduction to Industrial Networking*, Santa Clara, CA: Industrial Networking Inc., 1987.

3 ARCNET—A LAN FOR THE FACTORY

INTRODUCTION TO ARCNET

What is ARCNET? It is a high-performance, token-passing local area network developed by Datapoint Corporation in 1977 which has since grown to an installed user base exceeding 600,000 nodes worldwide. Today, ARCNET continues to thrive due to its reliability, economy and availability—promoted by a second-sourced chip set.

The term ARCNET is actually a combination of two acronyms—ARC, attached resource computer, and NET, network. Datapoint was a pioneer in bringing distributed data processing into the business environment. But customers soon found that they needed to share various computer resources while maintaining the benefits of distributed processing. ARCNET answered this need—supporting up to 255 nodes and transmitting data at 2.5 megabits/sec.

ARCNET is classified as a token bus system, although the physical topology is actually a distributed star when active hubs are used to interconnect the nodes on the network. In most installations, RG 62/u coaxial cable serves as the physical media. For specialized applications, ARCNET also supports twisted pair and fiber optic cabling.

ARCNET is a baseband network extending up to four miles when active hubs are used. The maximum distance allowed between nodes is 2000 feet.

User-Transparent Operation

The ARCNET token continually passes from node to node around the network. When a node receives the token, it becomes the momentary master of the network and transmits its message. After a node sends its message, active hubs relay the message to every node on the network.

ARCNET's token-passing scheme is deterministic. One can predict the maximum amount of time it will take for a node to receive the token. This guaranteed response time is especially critical in real-time processing environments such as industrial control systems.

ARCNET also supports variable length data packets, provides built-in error-checking and automatically acknowledges successful packet transmissions. Network control and protocol are transparent to the user relieving the application software this burdensome task.

ARCNET reconfigures the network automatically when nodes are added to or deleted from the network. During reconfiguration, the system determines the unique address of each node and the token passes from node to node without wasting tokens on non-existing nodes.

Use of Active Hubs

Active hubs interconnect nodes in the ARCNET network, providing isolation and cabling flexibility. Active hubs also retransmit messages sent by the communicating node to all the remaining nodes on the network—acting as an electronic repeater.

Active hubs isolate network nodes. A fault on one node or cable does not usually impact the remaining nodes because each node connects to a single port on an active hub. If a node fails, the network is automatically reconfigured and the network ignores the disabled node—an important feature for non-stop systems.

Network Interface Modules

Network interface modules (NIMs) interface ARCNET to a particular computer's bus structure. NIMs are available on all the popular bus structures and appear to the host computer as both a memory and an I/O port.

Each NIM incorporates an integral transceiver designed to support different cable media. Currently, transceivers exist for fiber optic, twisted pair and coaxial cabling.

ARCNETs DEVELOPMENT

In 1976, a group of four young, enterprising hardware and software designers from Datapoint Corporation embarked on a networking system project that evolved into one of the most widely-used but relatively unknown LANs in the industry—ARCNET.

The goal of this design team was to create an intelligent link between Datapoint's distributed data processing computer systems so that customers could share data and computer resources while still retaining the benefits of stand-alone processing.

The fathers of ARCNET had quite a few requirements for the system they were designing. They wanted the network to have high speed data transmission rates, minimal performance loss under increasing transmission loads, high reliability and media flexibility.

When choosing ARCNET's raw bit rate for data transfer, 2.5 megabits/second was selected primarily because that was the transfer rate of the disks Datapoint was using at the time.

To maintain steady system performance even under increasing load, ARCNET's designers studied how networks function in real life situations and came to the conclusion that more than 90 percent of all messages transmitted on a network are small messages. With that in mind, they designed ARCNET with the capability to move small messages fast.

ARCNET was also designed to be reliable. If one node on the network becomes disabled, the rest of the network continues to function. When a cable is broken, the LAN becomes two subnetworks that function independently.

Additionally, ARCNET was designed to be completely free from cabling selection restraints. Nodes link to the network via cabling that attaches to a network interface module (NIM), a card inserted into an expansion slot on each node in the network. The NIM interconnects the computer to the

ARCNET network, enabling the installer to attach any type of media—coaxial, twisted pair or fiber optic cable—to the computer. The other end of the cable then attaches to an active hub which meets electrical restrictions, such as cable termination, and retransmits the signal passed along the network to all nodes attached to the hub.

By fall of 1977, the ARCNET project was complete. Datapoint could now offer distributed processing systems that could share data and computer resources. However, ARCNET drew little fanfare or recognition, primarily because it wasn't developed as product in itself, but as an integrated part of Datapoint's computing machines.

Datapoint kept ARCNET as a proprietary network until 1982 when Datapoint allowed Standard Microsystems Corporation to market an ARCNET chip set to other OEMs, who in turn marketed the ARCNET technology as a proprietary network. Although this move by Datapoint expanded the use of the ARCNET technology, the proprietary LANs did little for ARCNET's name recognition. A movement is underway to popularize the ARCNET name.

CASE STUDIES: USING ARCNET IN THE FACTORY

Two case histories are included in this primer to provide a better understanding about the actual application of ARCNET on the factory floor. The first case study explains how Pro-Log Corporation of Monterey, California, uses ARCNET to link its CIM system. The second case study highlights how a Virginia-based maufacturer—Robertshaw Controls—implemented ARCNET in its automated testing system.

Computer Integrated Manufacturing at Pro-Log Corporation

Computer integrated manufacturing (CIM) is alive and well at Monterey, California-based Pro-Log Corporation. Within a period of two working years, this computer and board-level products manufacturer has designed and implemented an impressive manufacturing strategy that includes its own products as key components in the fully-integrated system.

The ARCNET Link

The ARCNET local area network (LAN) serves as the system's backbone, linking two Compaq file servers running NetWare System Fault Tolerant Level 2 software and 28 PCs on the plant floor, most of which are Pro-Log's own STD Bus, PC-compatible System 2 industrial computers. The computer-to-ARCNET link is provided through ARCNET S871P network interface modules manufactured by Contemporary Control Systems.

Pro-Log's CIM implementation provides access to manufacturing information company-wide. The file servers can retrieve data from the manufacturing database stored on Pro-Log's HP3000 minicomputer and the PCs, used for local processing and control, can also access the database.

Choosing a LAN

One of Pro-Log's biggest challenges in implementing this CIM system has been finding a LAN that could provide the reliability and deterministic qualities Pro-Log demands.

According to Russ Cunningham, senior systems programmer and coordinator of the CIM implementation project, "ARCNET is the third LAN we've used and we are very pleased with the reliability. ARCNET has been up and running for four months without a single problem."

With Pro-Log's first LAN implementation, roughly fifty percent of the network cards received from the vendor failed when they were plugged in. Pro-Log also experienced high network card failure rates with its second LAN implementation. The entire network went down weekly, completely shutting down the manufacturing operation until engineers resolved the problem.

Says Cunningham, "We are also very pleased with ARCNET's performance. We extensively tested the throughput of ARCNET and compared it to our previous LAN. Our tests showed that the raw bit rate didn't provide a complete measure of the LAN's performance. We compared a LAN running at 10 megabits/sec to ARCNET's 2.5 megabits/sec and found the 10 megabits/sec network's actual throughput to be 50 kilobytes/sec while ARCNET was 25 kilobytes/sec. We expected ARCNET to run four times slower based upon its raw bit rate."

It should be noted here that ARCNET is not responsible for the 25 kilobytes/sec throughput. This throughput rate is more a factor of the application software and the speed of the processor being used than the raw bit rate of ARCNET.

This point is further illustrated by one of ARCNET's designers, John Murphy who is now with Performance Technology, Inc. (San Antonio, Texas).

According to Mr. Murphy, "We currently run a 22-node ARCNET LAN that can move data between any two nodes at rates exceeding 125 kilobytes/sec."

Continued Murphy, "On very small networks, data rates as high as 200 kilobytes/sec are possible."

Pro-Log's CIM Strategy

Pro-Log's automated manufacturing system closely monitors every step in the manufacturing process from the receipt of components to the complete, fully-tested product. Even product design is integrated in the system. For example, board layouts are designed on a computer aided design (CAD) system and CAD files are transferred to the manufacturing building across the street via a serial link where they are downloaded into a System 2 and run through a custom conversion program. Data is then transferred via ARCNET into the automatic insertion machines on the plant floor.

The CIM process begins when a customer places an order. Purchase orders are entered into the HP3000 which in turn generates a list of components needed to fill the customer's request. Upon receipt of the components, each part is labeled for

bar code scanning. Then, receiving information (for example, vendor name and part number) is entered into the HP3000. Every two minutes, the HP3000 downloads this receiving information through the ARCNET LAN into computers on the plant floor.

Components make their way through the factory to one of several stocking locations depending upon how the parts will be used. ICs, for example, are stocked near the IC auto insertion machine.

The next step is running printed circuit boards through auto insertion. After this process, boards travel to the manual insertion area for special assembly where components such as connectors are loaded onto boards. Assembled boards then run through the wave soldering machine.

Extensive product testing follows board assembly. Boards are tested for solder shorts and missing components and then enter the secondary or touch up phase where defects are identified, scanned with the bar code scanner, and entered into the computer system via a touch screen computer system. After testing, boards undergo a 24-hour burn-in test and are subjected to functional tests performed with Pro-Log's System 2.

Pro-Log's CIM system has greatly enhanced the quality of the company's products while at the same time improving throughput time, decreasing work-in-progress, and lowering finished goods inventories.

Says Cunningham, "Our CIM system benefits everyone in the company, not only production. We are very close to

having a paperless factory that is not only more efficient, but opens up the flow of information throughout the entire company."

Robertshaw Controls Links Automated Testing System with ARCNET

ARCNET played an important role in linking the devices in an automated production test and calibration system for Robertshaw Controls of Richmond, Virginia. This manufacturer of industrial, automotive and appliance controls has many plants and product lines, each requiring unique testing.

Each of Robertshaw's plants has an Intel 310 Multibus host computer with 1 MB main memory, a 40 MB hard disk, a single floppy drive, and a 45 MB tape drive. Additionally, each 310 has a CRT and a printer and is equipped with a modem port to link remote plants with technical support personnel. Robertshaw uses the 310 to run application programs, generate management reports, and service download requests from stations. The 310 also enables engineers to query database information, and all of these functions can occur at the same time.

Testing stations are configured with STD Bus hardware from several manufacturers. Each STD system consists of an 8088 processor, an 8087 coprocessor, and a real-time operating system with data and program memory residing in battery-backed RAM. No mass storage resides at the individual stations. This eliminates the chance of an operator on the plant floor using an old version of a test program which could possibly invalidate a complete production run. The program memory is check-

summed between every test and upon failure, a download is requested from the host.

ARCNET provides the communications between the various test stations and the host computer. A Contemporary Control Systems' M871 Multibus ARCNET network interface module is installed in the host with Contemporary Control Systems' S871 STD Bus ARCNET modules installed in each station.

Robertshaw is enjoying tremendous increases in performance with its new automated testing LAN. Before this system was implemented, Robertshaw connected each station to the host via RS-232-C ports. The maximum data transfer rate per port was 1200 baud which translated into 30 minutes to transmit a 48K to 64K program from the host. Upon implementing the ARCNET LAN, these same programs required about 20 seconds to download from the host. Of these 20 seconds, over 19 seconds were consumed by the downloading software in the remote processor. Besides enhancing data speeds, the ARCNET LAN also provides error checking and protocol control which are built into the hardware.

Robertshaw's automated production test and calibration system streamlines product testing and provides central control over the testing programs used in each testing station. The system's high performance is directly linked to ARCNET's ability to transfer data at speeds far exceeding those attained with the previous data communications scheme with a much higher degree of accuracy.

Note: Additional information regarding the Robertshaw case history is available in the March, 1987, *Control Engineering*

article entitled: Board Level Computing via LAN Provides Automatic Testing.

OPENING UP THE ARCNET ARCHITECTURE

When it was originally developed, ARCNET was used solely by Datapoint. In 1981, Datapoint licensed the design of its ARCNET Controller chip set for manufacture to a third party—Standard Microsystems Corporation; Hauppauge, New York. This move opened up the ARCNET architecture prompting many networking companies to develop proprietary LANs using the ARCNET controller chip. By requiring this chip set in each ARCNET-based LAN, ARCNET product manufacturers could guarantee full compatibility with any ARCNET local area network.

One unfortunate trend began with the introduction of the chip set. Networking companies purchased the chip set and then marketed proprietary LANs. As a result, the ARCNET name received little recognition. Today, ARCNET is installed in over 600,000 nodes worldwide, but a surprising number of these users are not aware that they are using ARCNET.

Actions are underway to popularize the ARCNET name. Datapoint licensed the design of a CMOS-version of the ARCNET Controller chip set to NCR Corporation (Microelectronics Division; Colorado Springs, Colorado)—further extending the availability of ARCNET. Another positive move in the industry is ARCNET vendors' increasing efforts at promoting their networks as ARCNET LANs rather than as proprietary networks.

The ARCNET Trade Association

A trade association was also created to elevate ARCNET's awareness to potential and existing users, system designers and software and hardware equipment suppliers—the ARCNET Trade Association (ATA). This group was organized by ARCNET vendors to help educate network users about the advantages of ARCNET over other LAN technologies, and perhaps more importantly, to educate people about the benefits of using what has now become a de facto standard in the industry. Membership in the ATA is open to vendors, system integrators and users of ARCNET technology.

With tools such as the trade association, primers explaining the technology, and outreach to the press, perhaps ARCNET will continue to gain recognition and assume its rightful position in the LAN marketplace. ATA is located at: 3413 North Kennicott, Suite B; Arlington Heights, IL 60004 Tel. (312) 255-3003.

References

Thomas, George M. "Board Level Computing via LAN Provides Automatic Testing." *Control Engineering,* March 1987, pp. 190—195.

SECTION TWO:
ARCNET INTRICACIES

4 UNDERSTANDING THE ARCNET PROTOCOL

INTRODUCTION

ARCNET possesses a user-transparent protocol; that is, the protocol which controls the passing of data between the nodes is independent of the user's application. The user is unaware of the intricacies behind ARCNET transmissions.

Protocol control and handling are accomplished by the ARCNET Controller (AC) integrated circuit. The AC is a dedicated processor which handles all of the interaction between nodes. It is the power of this processor which makes ARCNET so simple to use. The loading and unloading of data buffers is handled by the programmer and the flow of data between the nodes on the network is transparent to the user.

This section describes ARCNET's protocol in detail including how nodes are addressed and set up on the network, the five types of ARCNET transmissions, how the token is passed, network configuration and reconfiguration, and the effect that deleting and adding nodes has on the network. This section also describes the procedure for transmitting data over the ARCNET network.

NODE ADDRESSING

The AC and its support circuitry are located on a network interface module (NIM) which interfaces a microcomputer's

bus structure to ARCNET. The NIM plugs into an expansion slot on the microcomputer. ARCNET NIMs exist for all the popular microcomputer buses.

The NIM and its microcomputer are considered to be a node on the ARCNET network and, therefore, must be assigned a unique address. Setting the address on a NIM is usually accomplished with either a set of jumpers or an eight-position DIP switch. The eight switches can select valid node addresses from 1 to 255. Address 0 cannot be used since it is reserved by ARCNET to signify the transmission of a broadcast message—a message directed to all nodes on the network. Although each node on the network must have a unique address, the particular address selected is unimportant as long as it is not 0 and is not the same address as another node.

LOGICAL RING

ARCNET is classified as a token bus system because all nodes on the network have equal access to a shared connection. To avoid communications contention, the ARCNET token passes between all nodes. This token provides equal access to every node on the network.

The token is always passed to the node with the next highest address. Once the token reaches the highest active node (the highest possible node being 255), that node passes the token to the lowest active node (the lowest possible node is 1). The token continues to pass from node to node in what is termed a logical ring—the token actually passes from one node to another in a logical pattern as shown on the following page in Figure 4.

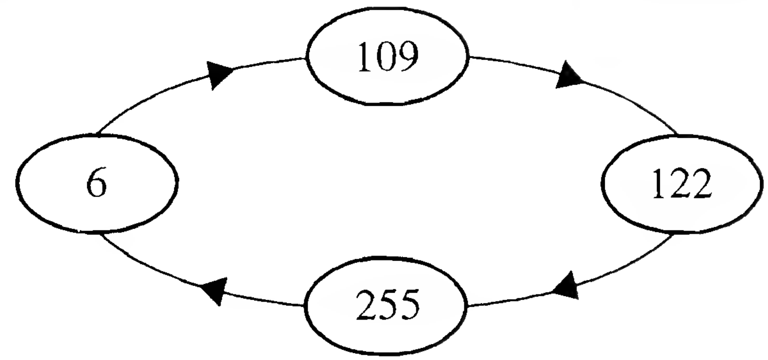

Figure 4. Symbolic Four-Node ARCNET Logical Ring

The logical ring has nothing to do with the physical placement of nodes, see Figure 5. The node with the next highest address is that node's logical neighbor. However, logical neighbors could be located at the extreme ends of a physical multinode network.

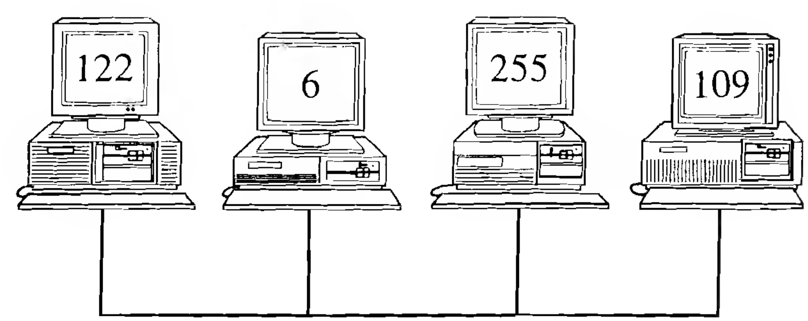

Figure 5. Four-Node Physical ARCNET LAN With Unique Node Addresses For The Same Four-Node Logical Network

During the ARCNET protocol discussion, all references will be made to the logical ring and not to the physical bus.

TRANSMISSION TYPES

The ARCNET protocol is comprised of five basic transmissions, including:

ITT - INVITATION TO TRANSMIT
FBE - FREE BUFFER ENQUIRY
PAC - PACKETS
ACK - ACKNOWLEDGMENT
NAK - NEGATIVE ACKNOWLEDGMENT

Each type of transmission is preceded by an alert burst which consists of six consecutive intervals of mark.

There are two basic, signal conditions that can occur over the ARCNET cable—a spacing condition and a mark condition. When no transmission occurs over the ARCNET network, the communications line idles in the spacing condition, logic 0. A logic 1, or mark condition, is represented by a dipulse being sent over the line. A dipulse is a combination of two pulses—a positive pulse immediately followed by a negative pulse. The time interval for each dipulse is 400 nanoseconds.

ARCNET is a character-oriented protocol (COP) in which each type of transmission is broken down into a series of eight-bit characters. Some of these characters conform to the American Standard for Communications Information Interchange (ASCII) and will be subsequently referred to by their ASCII symbol. However, before an eight-bit character is transmitted, ARCNET sends two intervals of mark and one interval

of space. For those familiar with asynchronous transmission, the two intervals of mark are similar to stop bits and the one space condition is similar to a start bit; therefore, any eight-bit character requires a total of eleven time intervals, three for the preamble and eight for the character itself.

ITT - Invitation To Transmit

The ITT grants network control to the node receiving it and is referred to as the token. The token is passed when one of two conditions occurs—after a node that is sending data completes its transmission and when a node receives the token and has no data to send. The node receiving the token then becomes the momentary master and can send a message.

Like all ARCNET transmissions, the ITT starts with an alert burst. The alert burst is then followed by three characters. The first of these three characters is an ASCII EOT (04 hex) which means end of transmission. The second and third characters are the DID or destination identification number of the node receiving the transmission. ARCNET can address 255 nodes, which requires eight bits of data to represent each node address. Therefore, only one eight-bit character is required to address any one of the 255 possible nodes. However, ARCNET repeats the DID and, therefore, requires two characters. The node with the designated DID is the node being granted control of the network, ultimately becoming the momentary master.

ITT =

ALERT	EOT	DID	DID

FBE - Free Buffer Enquiry

Before a destination node receives data from an originating node, the destination node must be queried to determine if it has a free data buffer to accept data—the destination node may have a full receive data buffer that has yet to be read by its host computer. Under these circumstances, the destination node is unavailable. The destination node must also be checked to verify that the receiver is enabled and that the node is on the network.

To check for an empty or available buffer at the destination node, the originating node transmits an FBE which consists of an alert burst followed by three characters. The first character is an ASCII ENQ (05 hex) which means enquiry. The second and third characters are the DID or destination identification address. Again, the DID is repeated requiring two characters. If the destination node sends a positive response, an ACK, the originating node can send a packet.

FBE =

ALERT	ENQ	DID	DID

PAC - Packets

Data is transferred between nodes in packets. Packet transmissions are more complex than the other four kinds of ARCNET transmissions because packet transmission length varies. ARCNET supports variable length packets and sends only the necessary characters that need to be transmitted. For example, in a fixed packet protocol, a message shorter than the packet size would have to be padded with null characters. These null characters "fill" the packets with the proper number of

characters. ARCNET has a minimum packet size of 1 character. There is a maximum packet length of 508 characters in ARCNET and only one packet may be sent before the transmitting node must pass the token. Messages longer than the maximum packet length require multiple transmissions for the entire message to be transmitted. Since the token is passed after every packet transmission, several round trips of the token over the logical ring will occur before the entire message is sent.

Packet transmissions begin with an alert burst followed by any number of characters from 8 to 516. The first character is an ASCII SOH (01 hex) which means start of header. The next character is an SID which is the source or originating node identification address. The SID can be one of 255 possible addresses and, therefore, can be represented by one eight-bit character. The DID follows and is repeated thus requiring two characters. After the second DID, ARCNET appends the continuation pointer (CP). The CP informs the destination node where in its packet memory it will find the beginning of the transmitted data. The CP facilitates the transmission of variable length packets since variable length packets occupy different locations in memory. After the CP is sent, data from a minimum of 1 to a maximum of 508 characters can be sent. The data characters are followed by two cyclic redundancy characters (CRC), forming a 16-bit word that verifies the integrity of the data in the packet. The proper CRC value is determined by the data pattern and is appended by the originating node. The destination node performs the same calculation and compares its result with the transmitted data's CRCs. The CRCs should match. If not, the data is corrupted and the

packet must be resent the next time the originating node receives the token.

PAC =

ALERT	SOH	SID	DID	DID	CP	DATA	...

DATA	CRC	CRC

ACK - Acknowledgment

The ACK transmission is a broadcast message that signifies a positive acknowledgment of either a PAC or an FBE. Transmission time is short since no node addresses are referenced. It is assumed that only the designated destination node will generate the ACK and since only one node, the source node, is looking for the reply, no node addressing is required. The ACK response either signifies that the receive buffer of the destination node is available or that a packet was received properly.

The ACK transmission starts with an alert burst and is followed by an ASCII ACK (06 hex).

ACK =

ALERT	ACK

NAK - Negative Acknowledgment

The other acknowledgment that can be generated is the NAK or negative acknowledgment. A NAK is sent by the destination node when its buffer is unavailable in response to receiving an FBE from the transmitting node. A NAK is not generated if a packet is not properly received.

The NAK transmission starts with an alert burst and is followed by an ASCII NAK (15 hex).

NAK =

ALERT	NAK

PASSING THE TOKEN

In discussing ARCNET's token passing protocol, assume that the network is functioning properly and that data transfers are not occurring. During this idle condition, the network is still very active passing tokens between nodes around the logical ring. When a node receives an ITT, the token, it checks to see if it has data to send. Realizing that it has no data to transfer, it passes the token by issuing an ITT to the next destination identification address (NID) which is the active node with the next highest address. Once the token is passed to the NID, the node receiving the token becomes the momentary master and upon realizing that no data transfer is required, passes the token by issuing an ITT to its NID.

For example, assume that four nodes are active on the network with node addresses 4, 57, 58 and 253. Node 4 will pass the token to node 57 since 57 is the next highest to 4. Node 57 passes to 58. Node 58 passes to 253 and 253 passes to node 4. Address 255 considers 1 as its closest node in the logical ring.

Each active node knows its NID since it stores the NID in a register in the AC and its NID is different from any other NID on the network. An active node needs only to know its logical neighbor, its NID, to successfully pass the token.

In this example, the NIDs are as follows:

ACTIVE NODE	NID
4	57
57	58
58	253
253	4

The token continues to pass from the lowest node address to the highest without referencing inactive node addresses.

CONFIGURATION AND RECONFIGURATION

Before the network can begin passing the token, the network must be configured. During configuration, ARCNET identifies all active nodes on the network. This identification process only has to occur once if the network continues to function properly. If a fault occurs, disrupting the proper passing of the token, the network must be reconfigured which again requires all the active nodes to identify themselves. Therefore, configuration and reconfiguration can be considered the same process.

Reconfiguring an ARCNET Network

If an active node fails to receive an ITT after 840 milliseconds or if a node is first powered up, the node will generate a reconfiguration burst (RECON). This RECON is a series of eight mark intervals followed by one space interval. This pattern is repeated 765 times. With ARCNET's 400 nanosecond time interval, the node will be active for 2754 microseconds which disrupts the successful passing of the token. The timing of this RECON burst exceeds the worst case transmission time and subsequent receipt of the next ITT. The node which should have received the token will not and, therefore, will not become the momentary master of the network. The node passing the token releases control of the network when it detects activity. This results in no nodes having the token. After no activity occurs for 78 microseconds, all nodes will realize that a reconfiguration is occurring. Each node then sets its NID to its own ID address and begins a time-out or calculates the following equation.

$$\text{TIME-OUT} = 146\ \mu s \times (255 - \text{ID})$$

Only the node with the highest address actually experiences a time-out. In fact, if the node was addressed at 255, no time-out will occur. The node that time-outs first sends an ITT to its NID and then listens for line activity. If no activity occurs after 74 microseconds, the sending node assumes that the node addressed by the NID either does not exist or is inactive. The sending node then increments its NID register and sends another ITT. Once the sending node detects line activity, it releases control of the line. The node addressed to the NID

will assume control of the network, generating some form of line activity within 74 µs.

This process continues until all active nodes in the network are determined. Once they are found, the token begins traveling around the network again. Note that during reconfiguration, all node addresses will be referenced. Once the active nodes are found, only those active node addresses will be referenced. No tokens will be wasted on non-existent nodes, greatly speeding up the process of passing the token. Reconfiguring a system takes from 24 to 61 milliseconds depending upon the number of active nodes and their absolute addresses.

Deleting a Node

As discussed before, reconfiguration occurs when a node is first added to the network or when active nodes sense no line activity after 78 µs.

Assume four nodes are active on the network with addresses 7, 18, 22 and 121. The token passes continually from 7 to 18 to 22 to 121 and then back to 7. Assume 18 drops from service and does not generate line activity within 74 µs after node 7 passed the token to node 18. Node 7 will detect this and assume that 18 is no longer active. Node 7 then increments its NID register and sends out another ITT. Node 7 continues to increment its NID register until line activity begins which will occur when node 22 receives the token. Node 7 then has its NID set to 22 and the token continues to pass from 22 to 121 to 7, excluding node 18. Therefore, deleting a node causes minimal disruption since a full reconfiguration is not invoked.

Adding a Node

Continuing with the example above, what will happen if node 18 returns to the network? Since node 18 is at first bypassed from token passing, it waits for 840 milliseconds and if it has not received the token, node 18 generates a reconfiguration burst that destroys the passing of the token. Once this occurs, ARCNET undergoes a full reconfiguration, reinstating node 18 to active status. After reconfiguration, the token again passes from node 7 to 18 to 22 to 121 and back to 7.

This process of reconfiguration demonstrates the flexibility of ARCNET. Adding or deleting nodes from the network is automatic and transparent to both the user and the network designer. This an important consideration in a factory network when the network integrity must not be impacted by a single node failing.

TRANSMITTING DATA

Assume that the network is properly configured and the token is passing freely between nodes. When a node receives the token, it has two courses of action, it can send a message or pass the token. Assume the node has a message to send.

Before sending a message, the originating node must verify that the destination node can receive a message. An FBE is sent to the destination node. If the destination node can receive a message, it returns an ACK. If it cannot accept the message, it sends a NAK.

Upon receipt of a NAK, the originating node simply passes the token and tries again the next time it receives the token. If

the originating node receives an ACK, it transmits a PAC. Upon receiving the PAC, the destination node verifies the message's integrity by comparing the CRC characters. If the message is acceptable, the destination node returns another ACK. Upon receipt of the ACK, the originating node passes the token. If the message was unacceptable, the destination node takes no further action. If the originating node receives no ACK within 74 microseconds, it assumes the message was lost and aborts the attempt. The token is then passed from the originating node which will reattempt a transmission the next time it receives the token.

The process is different for sending a broadcast message. When the originating node sends a broadcast message, no FBE is sent. Instead, the message is immediately sent to all nodes without generating either a positive or negative acknowledgment. The "handshake" protocol used to send normal messages is not implemented with broadcast messages. Nodes that are unable to receive messages will miss broadcast messages.

5 DETERMINING NETWORK RESPONSE TIME

INTRODUCTION

ARCNET is a deterministic network which means that one can predict the worst case delay possible when sending a message between nodes on the network. This is important to consider when designing a real-time plant floor system. This section will be helpful to those designers interested in the time calculations.

TRANSMISSION TIMES

ARCNET operates at 2.5 megabits per second and therefore the time interval for either a mark (logic 1) or a space (logic 0) is 400 nanoseconds. Determining a particular transmission time simply requires knowledge of the number of bits in the transmission.

All five transmission types begin with an alert burst (ALERT) which is comprised of 6 mark bits. The remainder of the transmission is organized in eight-bit characters; however, two marks and one space precede each character, requiring a total of 11 bits for a single character. To determine the transmission time, tally the number of characters and add an alert burst.

The following table provides the timing for each of ARCNET's five transmission types.

ITT =

ALERT	EOT	DID	DID

15.6 μs= 2.4 + 4.4 + 4.4 + 4.4

FBE =

ALERT	ENQ	DID	DID

15.6 μs = 2.4 + 4.4 + 4.4 + 4.4

ACK =

ALERT	ACK

6.8 μs = 2.4 + 4.4

NAK =

ALERT	NAK

6.8 μs = 2.4 + 4.4

PAC =

ALERT	SOH	SID	DID	DID	CP

...

33.2 μs + 4.4(n)μs = 2.4 + 4.4 + 4.4 +4.4 +4.4 + 4.4 +

DATA(n)	CRC	CRC

4.4(n) + 4.4 + 4.4

NETWORK DELAYS

The above times are for the five actual transmissions. However, there are delays inherent in both the physical media and the network interface modules that must be considered when calculating detailed timings. There are several delays, some of which are known and others which must be calculated on the basis of the application.

Tta - Turnaround Time = 12.6 μs

When an ARCNET NIM receives a transmission, it requires approximately 12.6 μs to respond to the transmission. After this turnaround delay, the NIM responds to the originating node.

Tpt - Token Propagation Time = 0 - 31 μs

A delay occurs between the sending of the token and the receipt of the same token by another node. This delay is called the token propagation time and it is generally a factor of the propagation time of the token through the cable. In other words, the token propagation time depends upon the type and length of the cable used. The delay should never exceed 31 μs. For worst case calculations, a value of 31 μs should be used even though Tpt will typically be at the low end of the 0 to 31 μs range.

Tpm - Message Propagation Time = 0 - 31 μs

The Message Propagation Time delay occurs between the node transmitting a message and the receiving node. This delay differs from the token propagation time in that the physi-

cal distance between a node passing the token to its next logical node can be different from the same originating node sending a message to an arbitrary node. Again, this delay ranges from 0 to 31 µs; therefore, when calculating worst case timings, a value of 31µs should be used.

Tbd - Broadcast Delay Time = 15.6 µs

Broadcast messages are different from other messages because no FBE is sent before a broadcast message and no ACK is generated after the message. However, a delay does occur after the node with the token transmits the broadcast message and passes the token. This delay is fixed at 15.6 µs.

Trp - Response Timeout = 74.6 µs

The maximum time that line activity can remain idle is 74.6 µs. The value is equal to twice the allowable message propagation time (Tpm) plus turnaround time (Tta) [= 2(31) + 12.6]. ARCNET allows a maximum of 31 µs of propagation delay and the turnaround delay is 12.6 µs.

In a typical example, after sending an ITT, the line becomes inactive. The ITT could propagate for 31µs to a distant node. Upon receiving the transmission, the distant node interprets it and generates a response which requires 12.6 µs. This response is then propagated back to the originating node taking an additional 31 µs. The originating node views the round trip response as no line activity for 74.6 µs. Therefore, in a properly operating network, line activity at any point should occur within 74.6 µs. If this does not occur, the originating node

interprets that the destination node is inactive and forces a reconfiguration.

Trc - Recovery Time = 3.4 µs

Sending an FBE to a destination node requires either an ACK (a positive response) or a NAK (a negative response). A PAC (a data packet) also requires an ACK. The time between a response and the subsequent start of the token pass is 3.4 µs.

Using the above information, a designer can predict worst case transmission times. In the following cases, timing is defined as the time between the start of a token pass to the start of the next token pass.

DETERMINING TOKEN PASS TIMINGS

Token Pass

In this case, the originating node passes the token to the next logical node.

ITT	15.6
Tta + Tpt	12.6 + Tpt
	28.2 + Tpt

Since Tpt can range from 0 to 31 µs, the token pass can take from 28.2 µs to 59.2 µs. The shorter the overall network distances, the faster the token pass.

Token Pass and Message

The timing for passing the token and a message depends upon the length of the message and the various propagation delays.

An example is calculated below.

ITT	15.6
Tta + Tpt	12.6 + Tpt
FBE	15.6
Tta + Tpm	12.6 + Tpm
ACK	6.8
Tta + Tpm	12.6 + Tpm
PAC	33.2 + 4.4(n)
Tta + Tpm	12.6 + Tpm
ACK	6.8
Tta + Tpm	12.6 + Tpm
	141.0 + 4.4(n) + Tpt + 4 (Tpm)

Assuming the maximum length message—508 characters—and the longest propagation time of 31 μs each for Tpt and Tpm, the token pass and message will occur in 2531.2 μs. Assuming a message of one character and zero propagation delays, the token pass and message will occur in 145.4 μs.

Token Pass and Message With Receiver Inhibited

Sometimes the destination node is unable to receive the message. Under these circumstances, the message is not sent and the token is passed.

A sample transmission time is calculated below.

ITT	15.6
Tta + Tpt	12.6 + Tpt
FBE	15.6
Tta + Tpm	12.6 + Tpm
NAK	6.8
Tta + Tpm	12.6 + Tpm
	75.8 + Tpt + 2 Tpm

By receiving the NAK, the message passing is aborted before it even occurs which minimizes time loss.

Token Pass and Broadcast Message

As mentioned previously, the broadcast message is unique in that ARCNET generates no FBE, ACK or NAK. The time required to send a broadcast message is important especially when transmitting critical messages such as alarm conditions.

The time duration of a broadcast message depends greatly on the length of the message.

ITT	15.6
Tta + Tpt	12.6 + Tpt
PAC	33.2 + 4.4(n)
Tbd	15.6
	77.0 + 4.4(n) + Tpt

Token Pass and Message With No Affirmative Response

This situation occurs when the receiving node finds an error with the message such as an invalid CRC check. Under these circumstances, no ACK is generated and the destination node experiences a timeout.

ITT	15.6
Tta + Tpt	12.6 + Tpt
FBE	15.6
Tta + Tpm	12.6 + Tpm
ACK	6.8
Tta + Tpm	12.6 + Tpm
PAC	33.2 + 4.4(n)
Trp	74.6
Trc	3.4
	187.0 + 4.4(n) + Tpt + 2 Tpm

Notice that the first ACK was received in response to the FBE; however, the message itself was not properly received. The originating node experiences a timeout and then recovers. Upon recovery, the originating node passes the token.

Token Pass and Message Sent To Inactive Destination

If an originating node inadvertently attempts to transmit a message to an inactive node, no response will be detected and a timeout will occur.

ITT	15.6
Tta + Tpt	12.6 + Tpt
FBE	15.6
Trp	74.6
Trc	3.4
	121.8 + Tpt

In this case, not even the first ACK was detected.

Token Pass With No Response

It is also possible that a node in the network may become inactive. This will not be detected until the token is passed to this node and no activity results. When this occurs, the originating node experiences a timeout, increments its NID,

and passes the token again.

ITT	15.6
Trp	74.6
Trc	3.4
	93.6

This condition does not cause a system reconfiguration since a node dropped from the network.

6 UNDERSTANDING NETWORK INTERFACE MODULES

INTRODUCTION

Interfacing ARCNET to microprocessor bus board products is accomplished with a network interface module (NIM). NIMs exist for all the popular bus structures such as Intel's Multibus I, VME, STD, PC and the Q bus. CCSI's P871 PC ARCNET NIM is pictured below in Figure 6. Frequently, terminology such as NIC, (network interface controller) and RIM (resource interface module) is used; however, these terms basically refer to the same functions—the interface of the host processor to ARCNET.

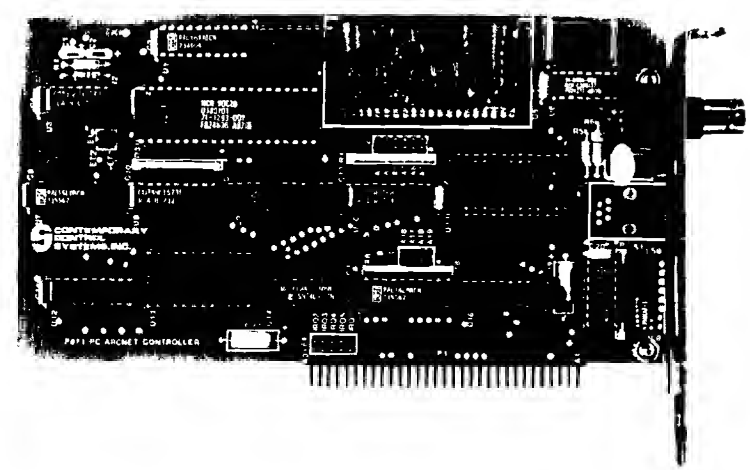

Figure 6. CCSI's P871 PC ARCNET NIM

Each NIM is comprised of four major components. The first component, the 9026 ARCNET controller integrated circuit (AC), is the heart of the NIM. Another component is the 9032 LANT which is the companion ARCNET transceiver. The third component, the LAND, is the cable transceiver or driver which varies according to the cable media being used. The last component on a NIM is the memory buffer which contains four packet buffers. In addition to the four major components, support components exist which address the unique requirements of the host processor. In this section, each major component of the NIM is studied.

ARCNET CONTROLLER

The AC contains a microcode sequencer, the microcode of which remains a trade secret of Datapoint Corporation. However, Datapoint Corporation licensed the technology to both Standard Microsystems Corporation and NCR Corporation, Microelectronics Division, and both companies have developed the AC using separate silicon processes. Still the AC pinouts are the same and the two units are interchangeable, thereby providing the industry with two sources of fully-compatible product. The AC requires connections to both the LANT and the memory buffer.

Although the AC is a 40-pin chip, it uses a multiplexed data and memory bus. The AC requires both memory addressing and input/output (I/O) addressing. Host processors that incorporate memory-mapped I/O bus structures need to allocate some portion of their memory space for the AC's I/O requirements. The AC requires two I/O ports and an address decoder which decodes these two I/O ports.

The AC operates from a 5 MHz clock and functions independently of the host processor's clock. Therefore, the host processor's access of NIM functions must be synchronized. The AC develops a signal called WAIT which suspends the host processor cycle so that the AC can process the host's command. This WAIT circuitry, located partly on the AC and partly on the NIM, is unique to the type of bus structure being supported. In general, the AC does not suspend the host during access of the NIM since transfers occur quickly.

The AC must know the NIM's node address which is usually set into DIP switches or jumpers. A parallel-to-serial shift register loads the node address serially into the AC using signals generated by the AC. The AC can also be programmed for extended distance operation by controlling the status of two input pins. One interrupt signal is also developed by the AC. This signal must be directed to the appropriate interrupt line on the NIM. The AC requires a significantly long (100 ms) reset signal. Usually this signal is generated by a one-shot circuit which stretches the power-on-reset signal from the host processor bus.

LANT

The LANT is a companion chip to the AC and works with the AC, decoding and encoding data from the network. A 20 MHz oscillator feeds the LANT. The LANT divides this signal and presents a 5 MHz signal to the AC. ARCNET uses asynchronous signaling, so the NIM must synchronize its receive circuitry with the incoming data. Several signal lines between the LANT and the AC interact to achieve this result which is data in an NRZ (non return to zero) format on the

AC's Rx input. A logic 1 for 400 ns represents a mark and a logic 0 for 400 ns represents a space. ARCNET transmits at a 2.5 MHz rate; therefore, each time interval is 400 ns long.

Transmit data from the AC is in a slightly different format than receive data. The Tx output from the AC drives the line to a logic 1 to indicate mark for only 200 ns and returns to a logic 0 for the remaining 200 ns. A space condition will remain as a logic 0 for 400 ns. This signal is fed into the LANT which in turn develops two signals called PULSE 1 and PULSE 2. During the 400 ns interval of mark, PULSE 1 remains on for the first 100 ns of the interval and is followed immediately by PULSE 2 which is asserted for another 100 ns. Both pulses are inactive during the remaining 200 ns of the interval. During a space interval, both PULSE 1 and PULSE 2 remain inactive. The two pulse signals modulate the cable transceiver or LAND and create what is called a dipulse.

Illustrations of Dipulse Signals Sent Over ARCNET Cable

Each of the following three illustrations has two graphs. The top graph represents the raw signals that flow over the cable in an ARCNET LAN. The bottom graph shows the signals that flow in or out of the AC and into the transceiver. The LSB, least significant bit, is sent first.

In the graph below, Figure 7, node 8 is passing the token to node F3. The graph shows an alert burst, 3 bits of preamble, an EOT (04 hex), 3 bits of preamble, a DID (F3 hex), 3 bits of preamble, and a DID (F3 hex).

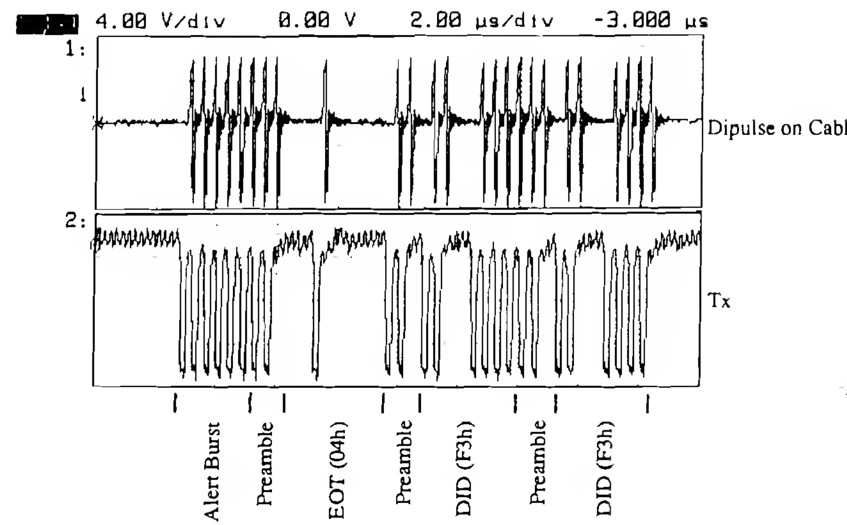

Figure 7. Token Transmit—Node 8 Passes Token to Node F3.

Since every node on ARCNET receives every message sent over the network, when node 8 sends the token to node F3, as shown in Figure 8 on the following page, it receives its own message but disregards it because the DID is not its own ID. Notice that the receive graphs differ from the transmit graphs in that the upper graph is further to the left than the lower graph. Also, the receive graphs are in the NRZ (non-return to zero) data format.

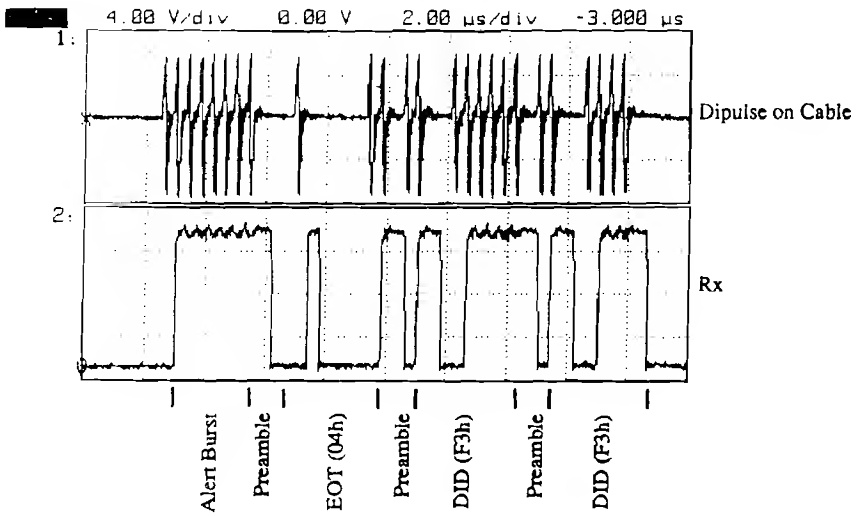

Figure 8. Token Receipt—Node 8 Passes Token to Node F3 and Receives Its Own Message

The graph on the next page, Figure 9, shows node 8 receiving a NAK message which is made up of an alert burst, 3 bits of preamble, and 1 data byte (15 hex).

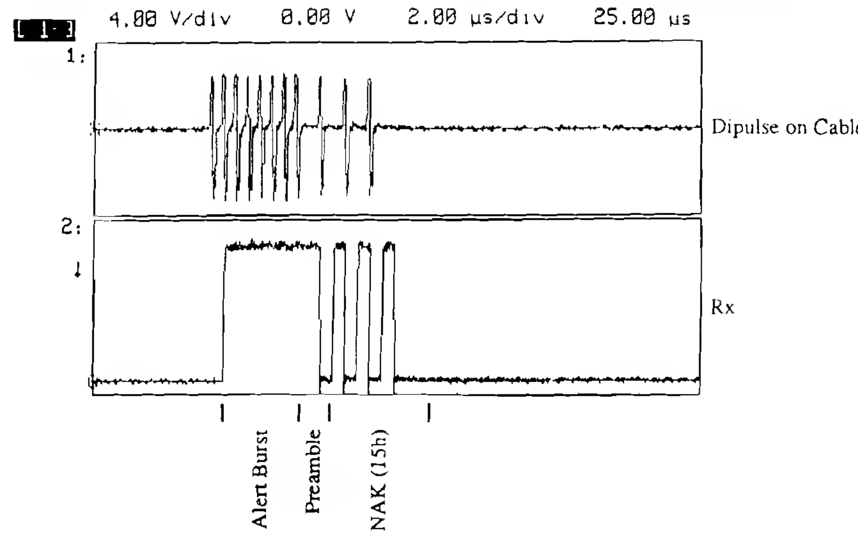

Figure 9. Node 8 Receives a NAK

LAND

ARCNET is currently supported on several different physical media such as coaxial cable, twisted pair and fiber optic cable. The choice of media dictates the choice of LAND. There are several popular cable transceivers and they all have the same footprint. Any of these transceivers can be installed on the NIM without modifying the NIM. LANDs usually are implemented with epoxy-coated hybrid technology and are very distinctive on the NIM.

7 PROGRAMMING THE NETWORK INTERFACE MODULE

INTRODUCTION

As mentioned before, the NIM is the device attached to the computer bus that contains the necessary circuitry to properly interface ARCNET with the host computer that commands the bus.

The intelligence of the NIM comes from the 9026 ARCNET Controller (AC) integrated circuit. Most of the programming of the NIM actually results in the programming of the AC located on the NIM. Therefore, frequent references will be made to the AC instead of the NIM.

The NIM does require both input/output (I/O) and memory (MEM) addressing by the host computer. Computers without I/O addressing must dedicate some portion of their memory to this I/O function.

During the discussion of the NIM, Contemporary Control Systems' S871P STD/PC ARCNET NIM, shown in Figure 10 on the next page, will be used as an example. Both memory and input/output mapping will vary on the different types of NIMs.

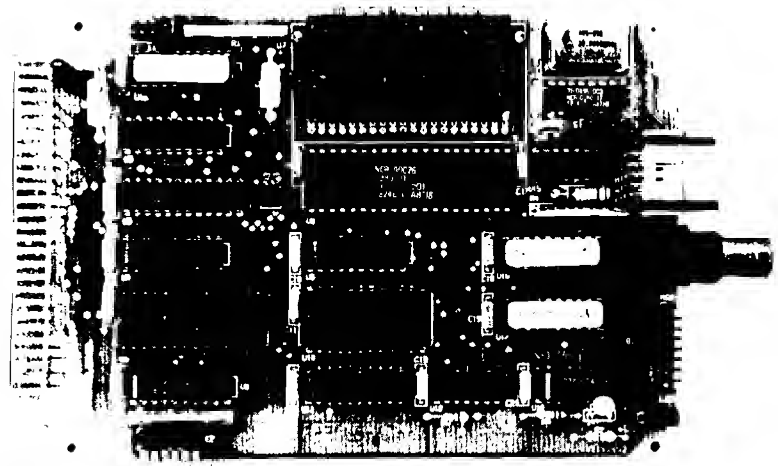

Figure 10. CCSI's S871P ARCNET Network Interface Module

To more fully illustrate the functional components of the S871P ARCNET Network Interface Module (NIM), a block diagram is included on the next page, Figure 11.

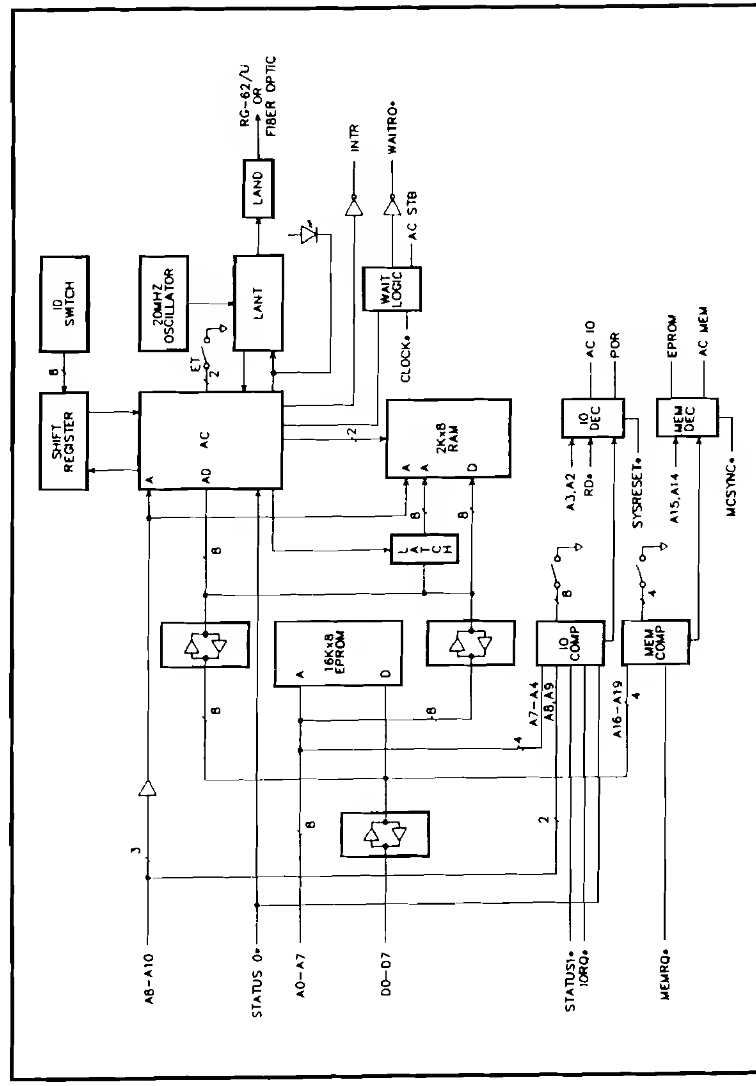

Figure 11. Block Diagram of CCSI's S871P ARCNET NIM

I/O ADDRESSING

The host exchanges control and status information with the NIM by reading and writing the command and status registers in the AC. In addition to these 9026 I/O commands, the S871P has commands which enable the user to control other portions of the NIM. The S871P maps into 16 consecutive I/O locations with the base residing on a 16-byte boundary. The I/O map is defined below.

I/O Address Map

I/O ADDRESS	READ/WRITE	FUNCTION
Base + 0	Read	9026 Status Register
Base + 0	Write	9026 Interrupt Register
Base + 1	Read	(Reserved)
Base + 1	Write	9026 Command Register
Base + 8	Write/Read	9026 Reset

COMMAND AND STATUS REGISTERS

9026 Status Register
Read at I/O Base + 0

The 9026 Status Register tracks the communications status of the host on the ARCNET local area network. The table below defines each bit in the 9026 Status Register.

9026 Status Register

BIT	NAME	FUNCTION
0	TA	**Transmitter Available.** When this bit equals "1," the node is available to transmit and any previous ENABLE TRANSMIT process has been completed.
1	TMA	**Transmit Message Acknowledged.** When this bit equals "1," a message sent from a previous ENABLE TRANSMIT command was acknowledged by the receiving node.
2	RECON	**Reconfiguration Flag.** When this bit equals "1," a system reconfiguration took place because an idle timeout expired. RECON is reset by the CLEAR FLAGS command.
3	TEST	**Test Flag.** This bit is for test and diagnostic purposes. Under normal operation, this bit equals "0."
4	POR	**Power On Reset.** When this bit equals "1," the AC has experienced a Power On Reset due to an active signal on the POR input. POR is reset by the CLEAR FLAGS command.
5	ETS1	**Extended Timeout Status 1.** The state of this bit reflects the logic level on the ETS1 pin. ETS1 will be "1" under normal operating conditions.
6	ETS2	**Extended Timeout Status 2.** Reflects the logic level on the ETS2 pin. Normally, this bit equals "1."
7	RI	**Receiver Inhibited.** When this bit equals "1," the AC is not accepting any messages from other nodes. If an ENABLE RECEIVE command has been issued to the AC, and no Power On Reset has occurred, the RI = "1" means that a message has been received into the RAM buffer over the network.

Note: If an AC experiences a Power On Reset, then the Status Register assumes the following states, where "x" indicates "same as previous state."

Bit:	7	6	5	4	3	2	1	0
State:	1	x	x	1	0	0	0	1

9026 Interrupt Register
Write At I/O Base + 0

The 9026 generates an interrupt signal on the INTR pin for most of the conditions that cause status bits to be set. A POR condition causes an unconditional (non-maskable) interrupt. A Mask Interrupt Register exists in which any combination of four other Status conditions may be set to cause interrupts. The four maskable status bits are grouped in the following table.

Maskable Status Bits

MASK BIT	STATUS CONDITION
7	Receive Inhibit
3	Test
2	Recon Timeout
0	Transmitter Available
6, 5, 4, 1	Not Used

Setting any of these four bits to a "1" will cause the INTR signal to become active whenever the corresponding status condition becomes true. The four other bits in the Mask Register should be considered as "don't care." Once the INTR signal is active (high), it can be deasserted by either resetting the corresponding Status Register bit or by resetting the Mask Register bit.

9026 Command Register
Write At I/O Base + 1

The 9026 Command Register issues transmission and reception commands to the COM9026. The commands are defined in the following table.

9026 C .mmand Register

DATA	FUNCTION
0000 0000	**RESERVED FOR FUTURE USE.**
0000 0001	**DISABLE TRANSMITTER.** Causes the AC to cancel its pending transmit command. This command also sets the TA bit the next time the AC receives the token.
0000 0010	**DISABLE RECEIVER.** Cancels a pending receive command. This command sets the RI bit the next time the AC receives the token. If a PACKET has already started arriving, then this command will have no effect.
000n n011	**ENABLE TRANSMIT from page nn.** Tells the AC to transmit out of RAM buffer page nn when it next receives the token. The TA and TMA bits are reset when the AC receives this command. The TA bit is set to logic "1" at completion of the transmission; the TMA bit may already be set at that time if the destination node has sent back an acknowledgement. If TA is not true, this command should not be issued.
b00n n100	**ENABLE RECEIVE to page nn.** Allows the AC to receive messages into RAM buffer page nn. This command sets the RI status bit to "0." If "b" in the command is "0," then only messages addressed to the AC's ID will be received. If "b" equals "1," then broadcast messages will also be accepted. RI is set by a successful message reception.
0000 s101	**DEFINE BUFFER SIZE.** Defines the size of an AC's RAM buffer. If bit "s" equals "1," the buffer is 2K bytes, and both short and long packets may be handled.
000r p110	**CLEAR FLAGS.** Resets the POR and/or the RECON status bits depending on the variable bits. If "r" is "1," then the RECON flag is cleared.

9026 Reset
Read or Write at I/O Base + 8

Any access to this address will cause the 110 msec pulse generator to reset the COM9026. A long (110 msec) reset pulse is necessary to reliably clear the 9026.

TRANSMITTING A PACKET

The NIM possesses four-512 byte page buffers available for ARCNET reception or transmission. The processor selects one of these four pages and writes the desired packet into the designated page buffer. Hexadecimal (base 16) data and address references will be used in the following examples.

In the first example, a short packet will be sent. Even though 1FF bytes are available in buffer memory, only FF bytes are used in a short packet transmission. Assume that node 3C intends to send the following ASCII message to node A2.

THIS IS A TEST

The destination node is A2 and therefore A2 is written into location 1 of the buffer memory. The last byte of the message is always located at address FF in a short packet message. The first byte of the message is 15 decimal locations before the last location since the message is 15 decimal bytes long. Converting to hexadecimal, the beginning of the message must therefore be located at address F1. The continuation pointer which is located at address 2 must be an F1. Also notice that the continuation pointer is actually the 2's complement of the number of data bytes in the packet with respect to 100.

NOTE: The first, and sometimes second, data byte(s) in every ARCNET message contain a system code. The system code allows more than one network operating system with unique message formats to operate over the same network. The box below describes how system code bytes are organized. System codes are assigned through the ARCNET Trade Association, (312) 369-2355. A system code of E9 is assumed in the following examples.

System Code Conventions

HEX CODE	DESCRIPTION
00 .. .. 7F	1-byte codes reserved by Datapoint.
8000 .. .. 80FF	2-byte codes to be used for diagnostic purposes.
8100 .. .. BFFF	2-byte codes that are assigned by the ARCNET Trade Association.
CO .. .. FF	1-byte codes that are assigned by the ARCNET Trade Association.

The short packet transmit buffer will look as follows below.

Short Packet Transmit Buffer

ADDRESS	DATA	COMMENT
00	.	(note 1)
01	A2	DID
02	F1	CP
.	.	
F1	E9	(System Code)
F2	54	T
F3	48	H
F4	49	I
F5	53	S
F6	20	(SP)
F7	49	I
F8	53	S
F9	20	(SP)
FA	41	A
FB	20	(SP)
FC	54	T
FD	45	E
FE	53	S
FF	54	T

Note 1: Regardless of what is written into location 0, the AC will always send the NIM ID.

If this message is sent to all nodes on the network, which is a broadcast message, then a 0 would be written into location 1.

After the buffer is loaded, the processor waits for the TA bit in the Status Register to become a logic 1, indicating that the transmitter is available because the previous packet has already been sent. Usually the TA bit is used in conjunction with an interrupt so that when the TA bit is set, the processor is automatically interrupted. While the processor is waiting, it can load a second packet in another packet buffer to prepare for a transmission. This is called double buffering of transmit data. The processor loads a buffer while a second buffer is being transmitted by the AC.

If the TA is set to a logic 1, an Enable Transmit command may be issued by specifying the buffer page where the buffer resides. Once the command is issued, both TA and TMA are reset and the AC does the rest without further software intervention.

Upon receipt of the token, the originating node sends the FBE. After receiving an ACK from the destination node, the PAC is sent. Upon a successful PAC transmission, the destination node sends another ACK which sets both TMA and TA. Setting of the TMA bit indicates a successful transmission. The originating node can then send another packet when it next receives the token.

It is possible that the transmission will be unsuccessful. Assume that for some reason the receiver at the destination node is not enabled. Under these circumstances, the packet will never be sent and the originating node's TA will never be set again. The programmer must provide a software timeout to guard against this occurrence. If a timeout occurs, indicating that TA has yet to be set, the programmer issues a Disable

Transmitter command which aborts the transmission and sets TA back to a logic 1 the next time the token is received.

If the originating node sends a packet to a nonexistent node, the packet will never be delivered. The TA bit will be set although the TMA will not. By testing the TMA bit, the programmer will know if the packet was successfully delivered.

If the Disable Transmitter command is issued and the TA bit fails to return to a logic 1 within the maximum time of a round trip of the token, the programmer can make any of the following conclusions.

- The originating node is disconnected from the network, preventing the receipt of the token.
- There are no remaining active nodes on the network to send a token.
- The receiving circuitry on the originating NIM is defective.

The programmer could use another software timeout to test for any of these occurrences.

RECEIVING A PACKET

To receive a packet, the processor must first enable the receiver and then specify a buffer page in memory to write the received message. The processor checks for the status of the RI bit after issuing a Receive command. When a packet is successfully received, the RI is set to a logic 1. Once the bit is a logic 1, the processor can read the page buffer. Usually an interrupt is associated with the RI being set to logic 1. This

action automatically signals to the processor that a message has been received successfully.

Assume that the destination node received the example message sent under the "Transmitting a Packet" section. The page buffer and the receiving node would look as follows.

Short Packet Receive Buffer

ADDRESS	DATA	COMMENT
00	3C	SID
01	A2	DID
02	F1	CP
.	.	.
F1	E9	(System Code)
F2	54	T
F3	48	H
F4	49	I
F5	53	S
F6	20	(SP)
F7	49	I
F8	53	S
F9	20	(SP)
FA	41	A
FB	20	(SP)
FC	54	T
FD	45	E
FE	53	S
FF	54	T

Notice that the receive message is almost identical to the transmit message except that location 0 contains the SID. The

transmitting node does not have to specify location 0 when it sends a message since the AC sends the local node address regardless of what is written in location 0. Therefore, the receive buffers have the SID always written into location 0.

If the message was a broadcast message, location 1 would have a zero. This can occur if the programmer enabled the receipt of broadcast messages.

If a receiver is disabled, the RI bit will go to a logic 1 the next time the node receives the token which will inhibit further reception of data. If a message was already in the process of being received, the process will complete itself independently of the disable command.

Receiving a Message Twice

Under certain conditions, it is possible for one node to receive the same message twice over ARCNET. For example, noise interference or a system reconfiguration could destroy an ACK being sent back to the Source ID. Upon unsuccessful receipt of the DIDs ACK, the Source ID will retransmit the message. If your network operating software does not protect you from the possibility of receiving a message twice, code should be written for this purpose.

8 CABLING ARCNET SYSTEMS

INTRODUCTION

This section addresses the cabling needs of ARCNET. Although ARCNET offers one of the most flexible cabling methods, wiring a factory floor can provide many challenges. Some areas may be hazardous or cabling may have to be run near generators of electrical noise. There are also significant cable maintenance considerations. The communications media must be reliable, and if a fault exists, it must be quickly isolated and repaired. ARCNET meets many of these needs by incorporating a variety of cabling options that can achieve almost any kind of cabling challenge.

PASSIVE HUBS

Connecting nodes within an ARCNET network is accomplished using hubs. A hub is simply a collection of wires in one central location. The concept behind hubs is simple; however, there are some design considerations that must be addressed when cabling ARCNET systems.

Each ARCNET node can be viewed as a voltage source in series with a 93 ohm resistor. To maximize power transfer and eliminate reflections, each node is terminated in its characteristic impedance—in this case 93 ohms. This termination can be accomplished by simply tying two nodes together.

Therefore, with point-to-point connections, as shown in Figure 12, no hub is required since both nodes in a two-node network are properly terminated. However, two-node networks are not that common.

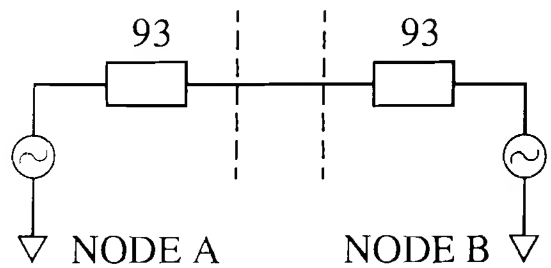

Figure 12. Point-to-Point Connection Without a Passive Hub

A three-node network requires more attention. If three nodes are directly tied together, the 93 ohm balance is lost since the driving point impedance (that impedance seen by any one node) is 46.5 ohms—the parallel combination of two 93 ohm resistors. To maintain the same driving point impedance of 93 ohms, series resistors must be added to each leg of the node as shown in Figure 13 on the next page.

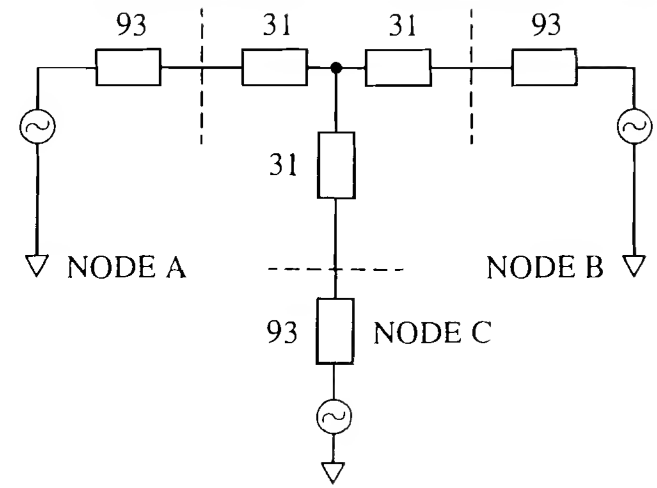

Figure 13. Three Nodes Connected With a Passive Hub

The value of 31 ohms for the added resistance can be calculated using circuit analysis. Therefore, a three-port passive hub, shown in Figure 14, can be constructed using three resistors.

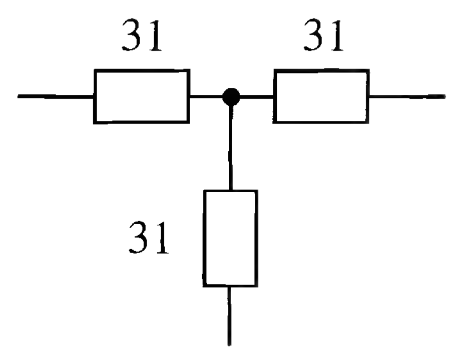

Figure 14. Three-Port Passive Hub

The same analysis can be carried further to a four-port hub, shown in Figure 15. In this case a 46.5 ohm resistor must be used. The same logic is used to extend the passive hub concept to many more ports.

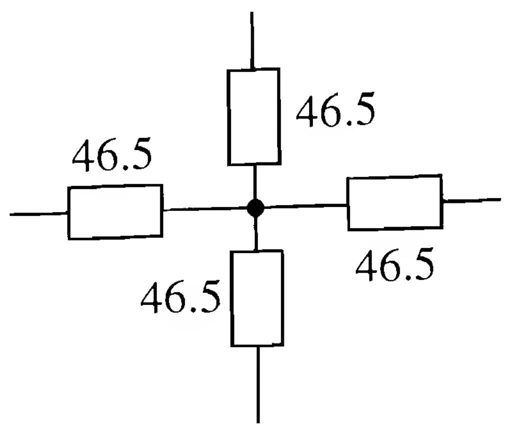

Figure 15. Four-Port Passive Hub

The use of series resistors will indeed achieve impedance matching; however, another problem is encountered. The voltage encountered by any node in a multinode network is much less than the voltage in a two-node network. This is because the series resistors also create a voltage divider attenuating the signal at the receiving node. Therefore, a four-port passive hub is the largest hub that can be used before experiencing unreliable operation.

Passive hubs have other shortcomings. Impedance balance is only maintained when all points on the hub are connected to a node. If three nodes are connected to a four-port hub, leaving the unused port open, impedance will become mismatched

which could disturb operations. Therefore, unused ports on a four-port hub should be terminated with a 93 ohm resistor. Four-port hubs are usually supplied with two terminating resistors just for the purpose of plugging into unused ports.

Another problem with passive hubs is that the series resistor not only attenuates the signal but also creates a low pass filter caused by the capacitance of the cable. This occurrence effectively limits the maximum length of passive hub wiring systems to approximately 100 to 200 feet. This cable length is hardly long enough, even for small plants, so passive hubs are usually restricted to few node, limited distance systems. A better alternative is provided by the active hub.

ACTIVE HUBS

The active hub, which is a hub that requires power to function, was introduced to alleviate the shortcomings of the passive hub. Features of an active hub are described below.

- Each port on the hub matches the output impedance of a node (93 ohms).
- Expansion beyond four ports is possible.
- Unused ports need not be terminated.
- A fault on one cable will not effect the integrity of the remaining cabling.
- Cable distances of 2,000 feet are possible between low impedance nodes.
- Hubs can be cascaded.

- Cables can be added to or deleted from the hub without imbalancing the network.
- Hubs serve as convenient patch panels for field wiring.

Active hubs incorporate electronic circuitry to achieve the functionality described above. Each point on the hub has the same transceiver as the one used on the NIM. Therefore, a four-port hub has four cable transceivers. The receive portion of the cable transceiver is enabled at all times while the transmitter portion is enabled by the hub electronics.

Assume a four-port hub connected to four active nodes A, B, C and D. Over an ARCNET network, only one node communicates at any one time. Since the hub is located electrically between all four nodes, it controls transmissions between all nodes. First, the hub "listens" for line activity from one of the nodes. As soon as activity is sensed, all transmitters, except the node generating the activity, are enabled. The hub receives the message from the transmitting node and retransmits the received message to all remaining nodes on the network acting like an electronic repeater. After activity ceases, the hub reverts back to an idle state, disabling all transmitters. As soon as activity begins again, all transmitters except the one receiving the activity are enabled.

Cable loading is not an issue with active hubs since a separate transceiver is dedicated to each active node. To the node, the hub appears as a two-node network. The active hub effectively isolates nodes from one another.

Since transceivers are used, the 2,000 foot transmission distance can be maintained between a node and a hub or between

two hubs. Hubs can be cascaded to the maximum length of an ARCNET system—4 miles.

Although ARCNET is a token bus system, the use of active hubs actually results in a distributed star topology. However, since hubs can be cascaded, the hub-to-hub connection appears as a bus. Therefore, the use of active hubs provides the benefits of both a star and bus topology. The distributed star topology is well suited to the factory environment. In general, equipment is clustered on the plant floor in numerically-controlled machining centers. In the office, workstations are also clustered. These clusters are best served with a single active hub located in the center of each cluster. Since the machining center and office pool are located at opposite ends of the plant, the two hubs must be connected with a "home run" connection. The only attention that must be paid during installation is cable length. Again, there can be only 2,000 feet of cable between a node and a hub or between two hubs. If this length is exceeded, another hub must be installed which would function as a repeater, see Figure 16.

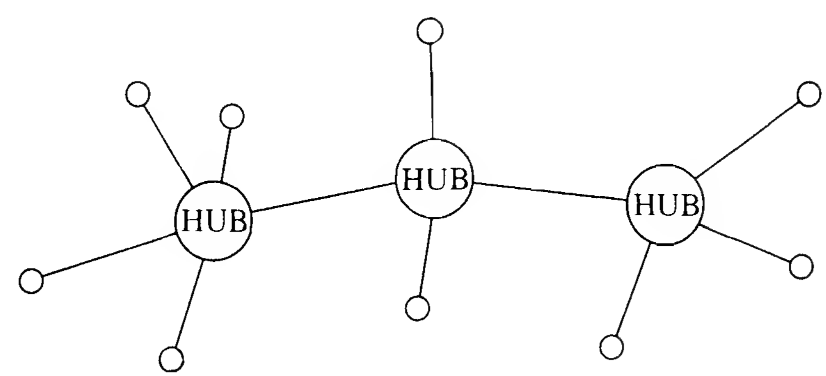

Figure 16. Distributed Star Topology Using Hubs

ALTERNATE CABLING SCHEMES

The above discussion refers to the use of RG 62/u coaxial cable which was originally specified for ARCNET communications. Coaxial cable is not the only means for communications, though, because the ARCNET controller chip (AC) does not pose significant restrictions upon the choice of physical media. The only restriction is that the delay of the signal through the media should not exceed 31 μs. With only this timing restriction, several other cabling approaches have evolved, some of which are suitable to factory floor communications. These alternate technologies also effect the 2000 feet maximum node-to-node specification.

Bus Topology

As mentioned before, using hubs with ARCNET creates a distributed star topology. A bus connection is more convenient since nodes can be connected together by simply linking all devices. However, this approach imposes severe impedance mismatches if conventional transceivers are used. Standard Microsystems Corporation (SMC) has developed alternate cable transceivers that have the same footprint as the conventional RG 62/u low impedance coaxial cable transceivers. Currently, two transceivers support a bus topology—the SMC 9058 High Impedance Transceiver and the SMC 9088 Twisted Pair Transceiver.

High Impedance Transceiver

The SMC 9058 High Impedance Transceiver (HIT) was designed to support bus connections of coaxial cable. The HIT replaces the conventional coaxial transceiver since it is the same physical size and uses the same signals as the conventional device. However, the HIT requires a -5 volt supply while the conventional device requires either a -5 volt or a -12 volt supply. The HIT also consumes significantly more power than the conventional device.

The HIT appears as a very high impedance to the cable allowing the parallel connection of several nodes. Impedance matching is achieved by terminating the cable at both ends of the bus with the cable's characteristic impedance. If RG 62/u coaxial cable is used, the terminating resistance is 93 ohms. An illustration of a bus connection using HIT transceivers is included below in Figure 17.

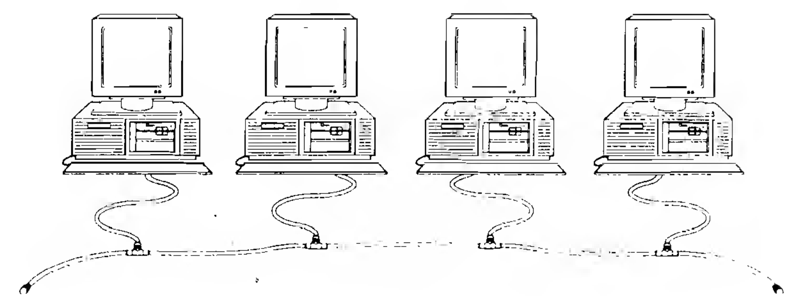

Figure 17. Bus or Multidrop Connection Using HIT Transceivers

Wiring a bus system is simple since conventional coaxial cable fixtures can be used such as tees. However, there are some restrictions with this approach. First, the maximum length of the bus connection cannot exceed 1000 feet with the further restriction that the maximum length of this bus

decreases inversely proportional to the number of nodes connected to the bus. The maximum number of nodes on the bus is 19. Under these circumstances the length of the bus cannot exceed 200 feet.

The HIT was designed to eliminate the need for using a hub. For small office installations, the HIT is attractive. However, the HIT also reintroduces the shortcomings of a bus topology in that a shorted cable impairs the complete system, rendering it useless. The short bus length may not be acceptable in a plant although it is possible to mix a bus system with a conventional system if the proper active hub is used. This type of hub is frequently called an active link and it is used to extend the length of bus systems.

Twisted Pair Transceiver

Another type of transceiver is the SMC9088 Twisted Pair Transceiver (TPT) which was was also developed by SMC. The TPT was designed to support the use of telephone cabling as a low cost, easy to install ARCNET media—perhaps even reusing the existing wiring of a building.

SMC developed the TPT to address these issues while at the same time achieving a multidrop connection. There are two telephone RJ 11 jacks located at each network interface module (NIM). Either jack can be used for inbound or outbound connection. The installation follows a daisy chain

connection of RJ 11 cabling. This approach is still a bus connection although it appears as a daisy chain, see Figure 18.

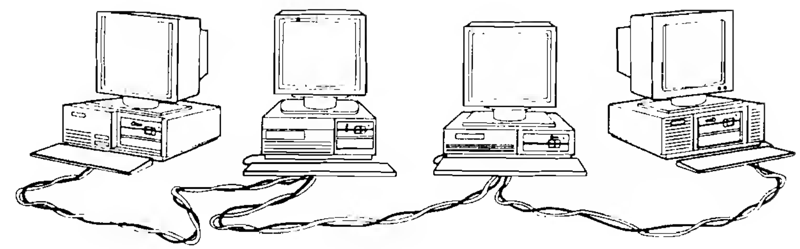

Figure 18. Wiring Four Nodes With Twisted Pair Cabling

Twisted Pair Bus Connection

Again, the network can be expanded using either active hubs or active links according to the following set of rules.

- The maximum length of each wire segment cannot exceed 400 feet.
- Up to ten TPT NIMs may be connected to each wire segment in a multidrop configuration.
- There must be a minimum of six feet of wire between each TPT NIM.
- Both ends of each wire segment must be properly terminated. If existing telephone lines are to be utilized:

Only one pair of wires from each phone bundle may be used for ARCNET data;

Only one pair of wires from each phone bundle may be used for a telephone.

- The topology will support either a 400 foot wire segment with four nodes or a 300 foot wire segment with ten nodes.
- A TPT network can be expanded using a twisted pair active link. This twisted pair technology is primarily intended to address the needs of the office; however, it may have applications in a plant environment.

Fiber Optics

Another type of cable technology is fiber optics. Raycom Systems of Boulder, Colorado, offers fiber optic cable transceivers that incorporate the same footprint as the low impedance LAND making fiber optics a simple option.

Fiber optics has many virtues. It has good transmission speed characteristics and very low error rates. It is immune to both radio frequency or electromagnetic interference. Since fiber optic cable does not radiate signals and cannot be easily tapped, it is well-suited for high security installations. Fiber optic cable is also immune to lightning and does not require lightning arrestors. Since no electrical current flows in the fiber cable, there is no spark hazard allowing the use of fiber cable in hazardous areas.

Fiber optic cable need not be run through conduits, trays or wireways and is frequently laid on top of drop ceilings. No special supports are needed for fiber cable making installation

easy. There are still building code restrictions that apply to fiber cable if this cable is run in HVAC plenums or ducts. Teflon-coated fiber cable is available; however, it is much more expensive than conventional fiber cable. Specialty cable is also available for outdoor installations making fiber optic cabling a flexible cabling alternative.

The Raycom transceivers allow a 4000 feet node-to-node connection, extending the 2000 feet constraint using coaxial cable. However, single fiber cable has a longer propagation delay than coaxial cable which means that the overall network length will be less than 4 miles. Regardless of the cable technology, the network designers must verify that no two nodes on the network will experience a delay greater than 31 µs. Fiber optics cabling is inherently a point-to-point connection. The use of hubs expands its effectiveness.

ACTIVE HUB DESIGNS

Active hubs are available from several manufacturers, many of whom are stressing their own cable solutions. Hubs can be classified as either fixed-port stand-alone, variable port modular, active link or board level. The fixed port units usually have eight ports since eight ports address the needs of a small office. To expand beyond eight ports, active hubs can be cascaded. Cascading is accomplished by connecting hubs together; however, this eliminates two ports from workstation usage. Two cascaded eight-port hubs have 14 ports available for workstation usage, not 16.

Modular Active Hubs

The variable port or modular active hub allows expansion usually in groups of four. There is no penalty for expansion since the active hub signaling electronics reside on a backplane. Adding four more workstations simply requires the insertion of a four-port secondary module. The modular hubs are available in either stand-alone or rack-mounted versions and are usually shipped with a single master module already installed. Expansion is accomplished with secondary modules. Secondary modules are also available with different cabling technologies. Thus, a user can mix fiber optic, low impedance coaxial cable, high impedance cable and twisted pair cabling in the same hub. This approach offers the most flexibility in cabling. Two modular active hubs are shown on the opposite page in Figure 19 below and Figure 20 on the following page.

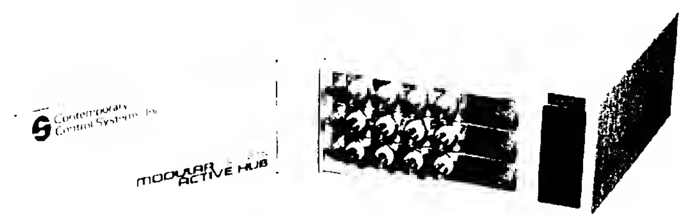

Figure 19. CCSI's 9003 Modular Active Hub

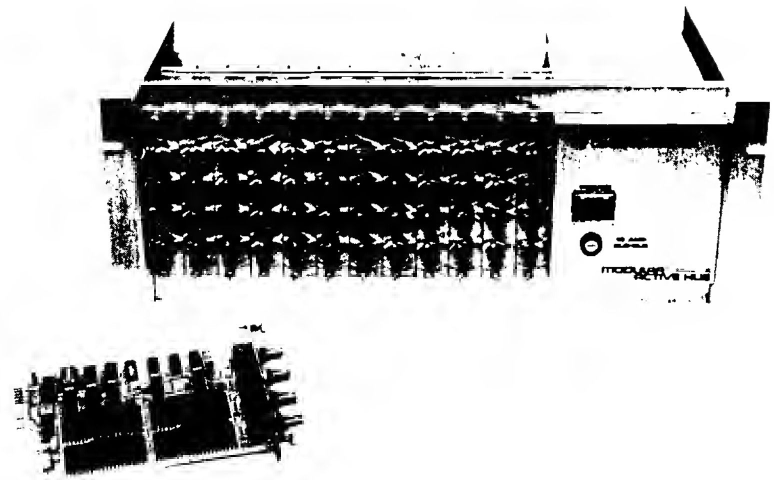

Figure 20. CCSI's 4012 Modular Rack-Mounted Active Hub

Active Links

The active link is generally used as a repeater or as an expander to a different cabling technology. For example, when the limitations of a bus connection are encountered, an active link is used to expand the network to another bus. If a coaxial cable connection to fiber optics is required, an appropriate active link is used. Active links are usually available as either two- or three-port devices.

Board-Level Active Hubs

Board-level active hubs are generally the least expensive active hub solution because they can tap into the power supply of the microprocessor bus, eliminating the need for a separate power supply. For example, assume that four PC Bus

computers are being networked and a hub is needed. A four port, board-level hub with the form factor of a PC card could be inserted in one of the networked PCs. A cable would then connect this board-level hub to the ARCNET NIM in the PC.

Installing a board-level hub is simple. Insert the module into an expansion slot in one of the microcomputers to be networked. Then, run cables from the hub to a maximum of three other nodes, connecting the cable to the network interface module in each node.

Using board-level active hubs, the number of available nodes can be expanded in two ways. To illustrate these methods, we will refer to CCSI's 771 Board-Level Active Hub, pictured in Figure 21 below.

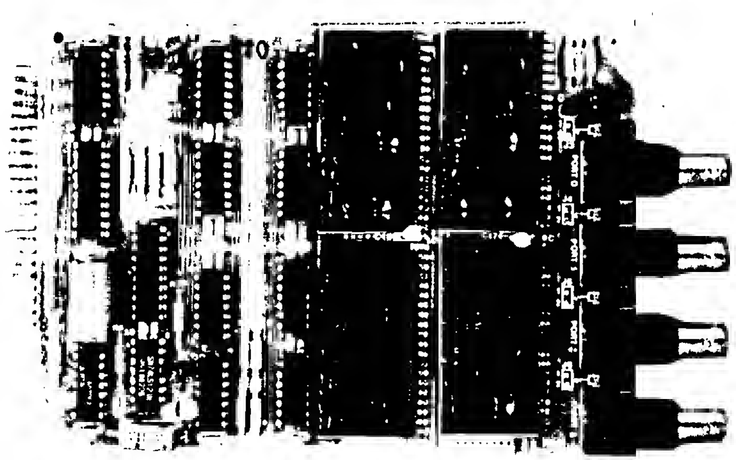

Figure 21. CCSI's 771 Board-Level Active Hub

The 771 Board-Level Active Hub, shown on the previous page, interconnects STD and STD/PC-compatible systems and is available as either a master or secondary module.

The first method for adding hub ports is to install a second 771 master module into the STD rack and connect the two 771 master modules with a short cable. This option offers the easiest expansion; however, one port is lost on each module to interconnect the two modules.

The second expansion option does not require the interconnection of master modules. In this configuration, one master module and up to nine secondary modules are installed in a separate STD Bus powered card cage dedicated for active hub use. A stand-alone card cage is required because bus signals are used to communicate between the master and secondary modules. This option accommodates up to forty active hub nodes.

Figure 22, shown on the following page, depicts a typical implementation of board-level active hubs—one card provides active hub functionality for up to four nodes.

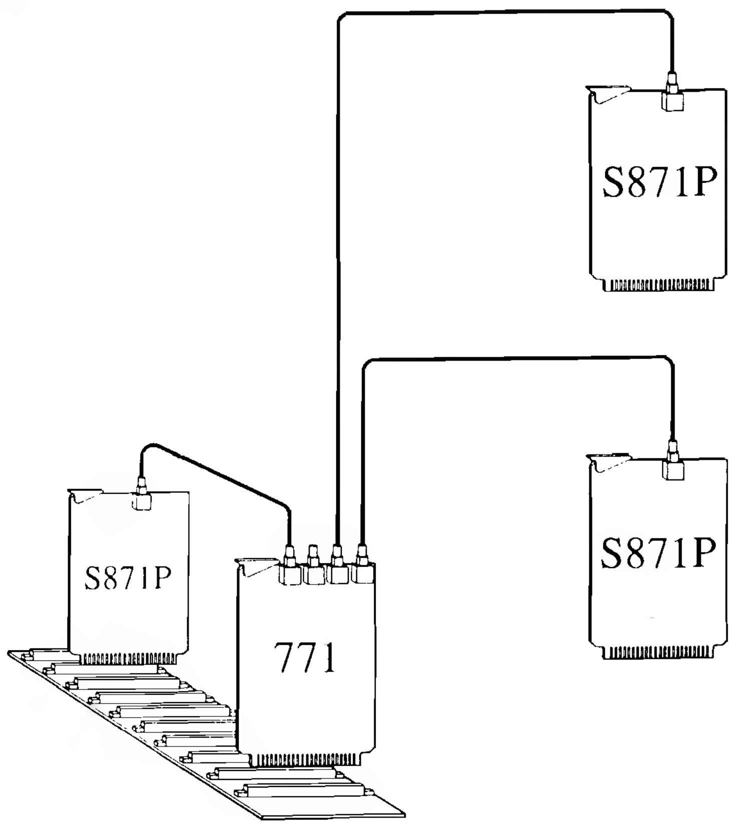

Figure 22. CCSI's 771 Board-Level Active Hub Connecting Three S871P NIMs

9 FINAL THOUGHTS

To elevate ARCNET's presence in the LAN marketplace, vendors, users and systems integrators must work together to develop a standard, open architecture system. Action toward achieving this goal is currently underway through groups such as the ARCNET Trade Association. This organization, which includes a membership of users, systems integrators and inventors, is considering plans to achieve IEEE certification for ARCNET which will fully standardize this protocol and Datapoint, the originator of ARCNET fully endorses this action.

According to Robert J. Potter, president and chief executive officer of Datapoint Corporation, "It is Datapoint's intent that the ARCNET be an open standard so that it will encompass a wide variety of attachments and accommodate a wide variety of protocols."

In the course of its work toward standardizing ARCNET, ATA is being confronted with many of the same issues that the MAP committee is facing in its development of an open architecture standard—the scope of this task is enormous.

ARCNET is an evolving technology and ATA is actively pursuing the possibility of enhancing the ARCNET protocol in areas such as data transmission speed. ATA is also working toward standardizing ARCNET to guarantee multivendor connectivity over the various types of physical media that

ARCNET supports. Any of these changes would maintain upward compatibility.

As LAN users, we have to keep an open mind toward existing de facto standards and encourage their use and adoption where appropriate. CCSI encourages comments from industry on how to improve ARCNET to meet the needs of industry and on how we can improve this primer to make it more useful. We welcome your input.

APPENDICES

Appendix A was written for this primer to provide assistance in developing transmit and receive routines for ARCNET. The following routines are written in 8086 Assembler.

APPENDIX A

SAMPLE ARCNET TRANSMIT AND RECEIVE ROUTINES

System Definitions Routine

```
;
;NIM and system equates
;
cr          equ     13
lf          equ     10
vendor_id   equ     0e9h                    ;vendor specific id code
;                                           ;this allows us to ignore
;                                           ;messages from different
;                                           ;vendor software
;
;IBM BIOS equates
;
bios_seg    equ     40h                     ;BIOS segment
timer_low   equ     6ch                     ;low word of timer
                                            ;count
timer_high  equ     6eh                     ;high word of timer
                                            ;count
;
;NIM I/O equates
;
iobase      equ     02e0h                   ;base I/O address of
                                            ;arcnet card
nimstatus   equ     iobase+0                ;read NIM status register
nimintmask  equ     iobase+0                ;write NIM interrupt
                                            ;mask
nimcommand equ      iobase+1                ;write NIM command
                                            ;register
                                            ;read undefined
nimreset    equ     iobase+8                ;read or write resets
                                            ;NIM
;
```

```
;NIM memory equates
;
membase       equ         0d000h              ;base address of arcnet
                                              ;packet buffer
;
;packet buffer mapping for short packets
;
packet_seg    equ         membase             ;we need this if we
                                              ;want to
                                              ;gen a com file
packet_buffer  SEGMENT at membase
source_id     DB          ?                   ;packet source ID
destination_id DB         ?                   ;packet destination ID
byte_count    DB          ?                   ;packet byte count
packet_data   DB          256-3 dup (?)       ;data buffer is 256
                                              ;bytes minus the
                                              ;previously defined 3
                                              ;bytes
packet_buffer ENDS
;
data          SEGMENT PARA   PUBLIC
data_seg      equ         data
kbdbuf        EQU         $                   ;keyboard message
                                              ;buffer
max_chars     DB          80                  ;max number of bytes
                                              ;per message
chars_entered DB          ?                   ;number of characters
                                              ;actually
                                              ;entered
msgbuf        DB          80 dup (?)          ;message buffer
table         DB          "0123456789ABCDEF"
msg1          DB          cr,lf,"Error NIM signature not found after reset","$"
msg2          DB          cr,lf,"Error incorrect status found after reset","$"
msg4          DB          cr,lf,"Error power on reset flag could not be cleared","$"
msg5          DB          cr,lf,"Warning no other nodes on network (token not found)","$"
msg7          DB          cr,lf,"Message sent","$"
msg8          DB          cr,lf,"Enter message to broadcast CR = none",cr,lf,"$"
msg9          DB          cr,lf,"Waiting for packet. Press any key to abort ",cr,lf,"$"
msg10         DB          cr,lf,"Transmitter not available for use","$"
data          ENDS
```


Main Routine

```
;
;             initialize NIM
;             check for other nodes on the network
;             send message
;             receive message
;             end
;
code          SEGMENT PARA   PUBLIC
              ASSUME  cs:code,ds:data,es:packet_buffer
;
start                 PROC    far
                      cld
                      mov     ax,data_seg        ;initialize segments
                      mov     ds,ax
                      mov     ax,packet_seg
                      mov     es,ax
                      call    initnim            ;initialize nim
                      call    chk_net_activ      ;check for other nodes
                                                 ;on the network
                      call    send_msg           ;send message

                      call    rcv_msg            ;receive message
                      mov     ah,4ch
                      int     21h                ;terminate
start                 ENDP
```

NIM Initialization Routine

```
;
;
;           reset NIM
;           wait at least 100ms for reset to take place
;           if ARCNET reset signature not found
;                   print signature not found msg
;                   abort
;           if status register does not have valid reset flags set/cleared
;                   print invalid status after reset
;                   abort
;           clear reset and reconfigure flags
;           if flags not cleared
;                   print flag error
;                   abort
;           set NIM mode to allow small and large packets
;           disable nim interrupts
;           end
;
initnim         PROC    near
                xor     al,al
                mov     source_id,al        ;zero source id before
                                            ;reset
                mov     dx,nimreset
                out     dx,al               ;write anything to reset
                                            ;nim
                mov     dx,ds               ;save ds
                mov     ax,bios_seg         ;get BIOS segment
                mov     ds,ax
                mov     ax,ds:timer_low     ;get timer_low
                add     ax,4+1              ;set wait for 220ms
wait1:          cmp     ds:timer_low,ax     ;check if time expired
                jne     wait1               ;wait for timer
                                            ;yes i should be doing
                                            ;a double
                                            ;word compare and jl,
                                            ;but this
                                            ;is only a sample
```



```
                                            ;program
                mov     ds,dx               ;restore ds
                mov     al,source_id        ;after reset, source id
                                            ;should
                cmp     al,0d1h             ;be set to d1
                mov     dx,OFFSET msg1      ;get error message
                jne     error               ;if error go inform user
                mov     dx,nimstatus
                in      al,dx               ;get status register
                and     al,10011011b        ;mask out invalid bits
                cmp     al,10010001b        ;verify against expected
                                            ;results
                mov     dx,OFFSET msg2      ;get error message
                jne     error               ;jump if error
                mov     al,00011110b        ;clear power on reset and
                mov     dx,nimcommand       ;recon flags
                out     dx,al
                mov     dx,nimstatus        ;check status reg to see if
                in      al,dx               ;por flag was cleared
                and     al,00010000b
                mov     dx,OFFSET msg4
                jnz     error               ;jump if por flag not
                                            ;cleared error
                mov     al,00001101b        ;support short and long
                                            ;packets
                out     dx,al
                xor     al,al
                mov     dx,nimintmask
                out     dx,al               ;disable interrupts
                ret
error:          mov     ah,09               ;print string
                int     21h
                or      al,0ffh             ;set not zero flag
                ret
initnim         ENDP
;
```

Routine for Determining Network Activity

```
;
;check for network activity
;This check is performed by enabling and immediately
;disabling the receiver. The receiver inhibit flag will
;be set when the token is received. Therefore if the RI flag
;is not set in 840ms then there is either a problem
;with the flag or no other nodes are on the network
;
;               enable receiver
;               disable receiver
;               wait enough time for token to make a complete trip
;               if receiver is not available
;                       print no other nodes on network
;               end
;
chk_net_activ           PROC    near
                        mov     al,04               ;enable receiver to
                                                    ;buffer page 0
                        mov     dx,nimcommand       ;get command register
                        out     dx,al               ;write command
                        mov     al,02               ;disable receiver
                        out     dx,al
                        mov     cx,ds               ;save ds
                        mov     ax,bios_seg         ;get BIOS segment
                        mov     ds,ax
                        mov     bx,ds:timer_low     ;get timer_low
                        add     bx,15+1             ;set wait for > 840ms
                        mov     dx,nimstatus        ;point to status register
chknetwait1:            in      al,dx
                        and     al,10000000b        ;test RI flag
                        jnz     end_activ_chk       ;if set then other nodes
                                                    ;exist
                        cmp     ds:timer_low,bx     ;check if time expired
                        jne     chknetwait1         ;jump if not
                        mov     ds,cx               ;restore ds
                        mov     dx,OFFSET msg5;get no network activity
                                                    ;warning
```



```
                        mov     ah,09
                        int     21h
end_activ_chk:          mov     ds,cx               ;restore ds if ri was
                                                    ;found
                        ret
chk_net_activ           ENDP
;
```

Transmit Message Routine

```
;
;get and send message from console using packet buffer 0
;
;please note vendor specific ID support
;
;               if transmitter is not available
;                       print error msg
;                       abort
;               get message to send
;               if message length = 0
;                       abort
;               send message
;               inform user message sent
;               end
;
send_msg                PROC    near
                        mov     dx,nimstatus        ;check if transmitter
                                                    ;available
                        in      al,dx
                        and     al,00000001
                        jnz     send1               ;jump if available
                        mov     dx,OFFSET msg10
                        mov     ah,09               ;inform user transmitter
                        int     21h                 ;not available
                        jmp     sendend             ;abort
;
send1:                  mov     dx,OFFSET msg8;prompt user for input
                        mov     ah,09
                        int     21h
                        mov     dx,OFFSET kbdbuf
                        mov     ah,0ch              ;flush buffer and
                        mov     al,0ah              ;fill keyboard buffer
                        int     21h                 ;do it
                        mov     al,chars_entered
                        or      al,al
                        jz      sendend             ;abort if no message
;
```



```
                        mov     ah,0                ;clear high byte
                        mov     cx,ax               ;save byte count
                        inc     al                  ;adjust byte count for
                                                    ;vendor id
                        neg     al                  ;get twos cpl
                        mov     byte_count,al       ;set byte count
                        mov     di,ax               ;set destination
                        mov     BYTE PTR es: [di], vendor_ID  ;set
                                                    ;vendor id in buffer
                        inc     di                  ;walk over vendor id
                        mov     destination_id,0    ;broadcast packet
                        mov     si,OFFSET msgbuf;get start of message
                        repnz   movsb               ;copy msg to packet
                                                    ;buffer
;
                        mov     dx,nimcommand
                        mov     al,03               ;xmit from page 0
                        out     dx,al
                        mov     dx,OFFSET msg7 ;tell user message sent
                        mov     ah,09
                        int     21h
sendend:                ret
send_msg                ENDP
;
```

Receive Message Routine

```
;
;this routine will enable the receiver to ram page 0
;upon reception the data is displayed on the screen
;
;please note vendor specific ID support
;
;               do while no user abort
;                       if message received
;                       print message
;               end
;
rcv_msg                 PROC    near
                        mov     dx,OFFSET msg9 ;inform user of what
                                               ;we're up to
                        mov     ah,09
                        int     21h
rcv0:                   mov     dx,nimcommand
                        mov     al,84h          ;enable rcv to buffer 0
                        out     dx,al
                        mov     dx,nimstatus
rcvwait:                mov     ah,0bh          ;check for user abort
                        int     21h
                        cmp     al,0ffh
                        je      rcvend          ;abort if user hit any
                                                ;key
                        in      al,dx           ;check status port
                        and     al,80h          ;packet received?
                        jz      rcvwait         ;continue to wait if not
;
                        mov     al,byte_count   ;get received byte
                                                ;count-256
                        mov     ah,0
                        mov     si,ax
                        neg     al              ;un twos cpl count
                        mov     cx,ax           ;use byte count
;
```



```
                        mov     dl,es:[si]      ;get vendor ID from
                                                ;data packet
                        cmp     dl,vendor_id    ;check if ours
                        jne     rcv0            ;if not, ignore packet
                        inc     si              ;walk over vendor id
                        dec     cx              ;update byte count
;
                        mov     ah,02           ;MSDOS display
                                                ;character
rcvl1:                  mov     dl,es:[si]      ;get character
                        int     21h             ;print it
                        inc     si              ;bump pointer
                        loop    rcvl1           ;print entire string
rcvend:                 ret
rcv_msg                 ENDP
;
code                    ENDS
;
                        END     start
```

GLOSSARY

access method A method for determining the order in which several networked stations will share the transmission medium.

acknowledgment (ACK) An ARCNET transmission message sent to acknowledge either successful receipt of a data packet or to acknowledge that the destination node has a free data buffer.

active hubs Electronic devices that have two primary functions on an ARCNET network—retransmitting messages to every node on the network and providing electrical isolation for network nodes so that a fault on one node or cable does not impact the remaining nodes. Each network node connects directly to a single port on an active hub.

amplifier An electronic device that is placed in specific locations throughout a network that has the primary function of boosting electronic signal strength which weakens as signals pass through the network cable.

analog signal An electrical signal that is soley dependent upon magnitude to express information content.

ARCNET The 2.5 megabit/sec., token-passing protocol developed by Datapoint Corporation. The term ARCNET is a combination of two acronyms—ARC, attached resource computer, and NET, network.

ARCNET controller A dedicated processor which handles all of the interaction between nodes.

ASCII—American Standard Code for Information Interchange A 7-bit code for the exchange of information, particularly between communication devices. ASCII is the standard code used in most of today's microcomputers.

asynchronous transmission A serial transmission method where data is read at regular intervals by the receiving node without clocking information being transmitted. Bytes are made up of a single character, preceded by a start bit and usually a stop bit.

attenuation Loss of power or signal energy that occurs during transmission over communication lines, equipment or other devices.

baseband network The transmission of either analog or digital signals over the medium, without modulating a carrier. In a baseband network, one message is sent at a time over the network.

baud rate A unit of measure used to express the speed (bits per second) at which serial data is sent and received, often via a modem.

broadband network A network where signals are modulated into noninterfering frequencies before being sent over the transmission medium. This modulation allows many signals to pass over the transmission medium at one time.

broadcast medium A transmission system in which all messages are heard by all stations.

broadcast message A message addressed to all nodes of a local area network.

bus topology A local area network topology in which all stations attach to a single transmission medium—all stations hear all messages sent over the network.

cable termination Matching cable impedance with load impedance to attain maximum power transfer and to prevent reflections.

cable transceivers A combination transmitter/receiver that drives the network medium.

carrierband network This network requires modulation of signals before they enter the transmission medium.

carrier-sense multiple access with collision detection (CSMA/CD) A contention access method which allows multiple nodes on the network to share a single channel. Each node can sense when activity occurs on the network and waits to transmit until the channel is available. These networks can also detect collisions.

central controller On a LAN with a star topology, a central hub controller is used to control the flow of data between the nodes on the network.

character oriented protocol (COP) In the ARCNET LAN, each type of transmission is broken down into a series of eight-bit characters.

coaxial cable A type of electrical cable in which a piece of copper wire is surrounded by insulation and then surrounded by a tubular piece of metal mesh. In general, coaxial cable supports moderate to high data transmission speeds—1 to 15 megabits/second.

command register A register on the ARCNET controller (chip) which issues transmission and reception commands.

computer integrated manufacturing (CIM) A manufacturing strategy that brings together automated equipment and computers to fully automate the manufacturing process and the flow of information on the factory floor.

contention An access method where each node must compete for access to the network.

continuation pointer (CP) Informs the destination node where in its packet memory it will find the beginning of the transmitted data.

cyclic redundancy characters (CRC) A calculation which verifies the integrity of data in a data packet.

data packet A grouping of data to form a part of a message.

demodulate Deriving the original signal that was previously modulated.

deterministic network A network in which the worst case delay time for sending a message between nodes can be predicted.

digital signal An electrical signal transmitted as ones and zeros.

distributed star Interconnecting a cluster or grouping of equipment using a single active hub in the center of each cluster.

electronic repeaters Electronic devices which retransmit received signals.

fiber optic cabling A cable consisting of optical fiber in which light carries the information. This cabling offers the ultimate protection from electrical noise and offers high speed transmission rates—up to 200 megabits/second.

free buffer enquiry (FBE) Used to enquire if the receiving node has a free data buffer.

impedance The total opposition offered by an electrical circuit to the flow of an alternating current of a single frequency.

impedance mismatch A situation where impedances are different causing reflection of the signal resulting in loss of power transfer.

invitation to transmit (ITT) The token in an ARCNET LAN. The node possessing the token is the momentary master of the network.

I/O address An address that appears in the I/O address space of a computer.

LAND See cable transceiver.

LANT A companion chip to the ARCNET controller (AC).

local area network (LAN) A combination of hardware and software that enables two or more computerized devices to share data base information, hardware resources and other application programs.

logical ring In an ARCNET network, the token passes from one node to another in a logical pattern; the node with the next highest address receives the token. The logical ring has nothing to do with the physical placement of nodes.

Manufacturing Automation Protocol (MAP) A communications protocol developed by General Motors Corporation that is intended to serve as a standard that could be adopted by computer manufacturers.

mark A signaling condition indicating a logical one.

mask interrupt register A register in the ARCNET controller in which setting a particular combination of status conditions will cause interrupts on the host.

medium A person, mechanism, electronic pathway, or other means of conveying information from one point to another.

megabits per second A measurement of one million bits per second.

modem A device that modulates digital signals to analog signals and vice versa.

multidrop configuration A bus scheme which connects several nodes to a single cable, or bus, via a line tap.

negative acknowledgment (NAK) An ARCNET message indicating that the destination node does not have a free buffer—sent in response to an FBE.

network interface modules (NIMs) Printed circuit boards which interface the ARCNET network to a particular computer's bus structure.

network operating system software Software which uses a network for communication—interpreting of the information sent and received.

nodes Points in a network where service is provided, service is used, or communications channels are interconnected.

packet buffer The memory where packets reside.

packets (PAC) An information subset of a message.

passive hub A device which splits the cabling on a local area network using resistors to match impedance.

peer-to-peer network A network in which intelligent devices communicate directly with one another without relying on the host.

point-to-point data communication Connecting only two computerized devices for the sharing of data.

polling This is an access method where each node on the network is "asked" if it has anything to transmit.

programmable logic controllers (PLCs) A computerized device that runs automated equipment based upon software stored in the device.

protocol A set of rules that governs how communications actually is carried out over the network.

reconfiguration A change in the quantity, types or arrangement of the equipment connected to a computer system, or a change in the arrangement of connections in a network.

reconfiguration burst (RECON) In ARCNET, a RECON is a series of eight mark intervals followed by one space interval. The RECON pattern is repeated 765 times which destroys the token and causes network reconfiguration.

ring topology Nodes connect to one another forming one continuous loop.

RS-232-C An Electronic Industries Association serial communications standard.

shared resource network A network where resources such as fileservers, printers and databases are shared by all nodes.

SID The source or originating node identification address in an ARCNET system.

signaling method Method in which data is transmitted such as baseband, carrierband or broadband.

software drivers Host-dependent software which interfaces applications software to specific hardware.

space A signaling condition indicating a logical zero.

spacing condition No transmission is occurring; therefore, the communication line is in an idle state.

star topology A local area network topology that resembles a starfish because all nodes are wired outward from a central station or hub that provides cabling connection for each node.

status register A register in the ARCNET controller that tracks the communications status of the node on the ARCNET local area network.

timeout In an ARCNET network, after a defined period of time expires, the AC performs a specific task. For example, after 78 μs occurs and no network activity is sensed, the network undergoes reconfiguration.

token In an ARCNET local area network, the token is a unique combination of bits which grants a node the permission to transmit.

token passing system An access method where nodes must receive the token in sequence. The node with the token becomes the momentary master of the network.

topology The actual layout of the cables connecting the nodes to the LAN.

transmission media Anything such as wire, coaxial cable, fiber optics, air or a vacuum that is used to propagate an electrical signal.

twisted pair cabling Two insulated wires twisted in a uniform fashion so that each is equally exposed to electrical signals impinging upon the wires from their environment.

WAIT A command that suspends the host processor cycle so that the ARCNET controller can properly process the host's command.